

**PENGEMBANGAN SISTEM INFORMASI BERBASIS WEB
PENJURIAN ONLINE UNTUK PENGELOLAAN DATA KEGIATAN
PUPUK KALTIM INOVATION AWARD PADA DEPARTEMENT
INOVASI DAN PENGEMBANGAN MANAJEMEN PT. PUPUK
KALIMANTAN TIMUR**

SKRIPSI

Untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun oleh:

Farah Shofiatul Ummah

NIM: 145150207111039



**PROGRAM STUDI TEKNIK INFORMATIKA
JURUSAN TEKNIK INFORMATIKA
FAKULTAS ILMU KOMPUTER
UNIVERSITAS BRAWIJAYA
MALANG
2018**

PENGESAHAN

PENGEMBANGAN SISTEM INFORMASI BERBASIS WEB PENJURIAN ONLINE UNTUK
PENGELOLAAN DATA KEGIATAN PUPUK KALTIM INOVATION AWARD PADA
DEPARTEMEN INOVASI DAN PENGEMBANGAN MANAJEMEN PT. PUPUK
KALIMANTAN TIMUR

SKRIPSI

Diajukan untuk memenuhi sebagian persyaratan
memperoleh gelar Sarjana Komputer

Disusun Oleh:

Farah Shofiatul Ummah

NIM: 145150207111039

Skripsi ini telah diuji dan dinyatakan lulus pada

04 Mei 2018

Telah diperiksa dan disetujui oleh:

Dosen Pembimbing I



Fajar Pradana, S.ST., M.Eng.
NIP : 19871121 201504 1 004

Dosen Pembimbing II



Adam Hendra Brata, S.Kom., M.T., M.Sc.
NIK: 2016079 00105 1 001

Mengetahui

Ketua Jurusan Teknik Informatika



Astoto Kurniawan, S.T, M.T, Ph.D
NIP: 19710518 200312 1 001

PERNYATAAN ORISINALITAS

Saya menyatakan dengan sebenar-benarnya bahwa sepanjang pengetahuan saya, di dalam naskah skripsi ini tidak terdapat karya ilmiah yang pernah diajukan oleh orang lain untuk memperoleh gelar akademik di suatu perguruan tinggi, dan tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis disitasi dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila ternyata didalam naskah skripsi ini dapat dibuktikan terdapat unsur-unsur plagiasi, saya bersedia skripsi ini digugurkan dan gelar akademik yang telah saya peroleh (sarjana) dibatalkan, serta diproses sesuai dengan peraturan perundang-undangan yang berlaku (UU No. 20 Tahun 2003, Pasal 25 ayat 2 dan Pasal 70).

Malang, 23 Maret 2018



Farah Shofiatul Ummah

NIM: 145150207111039

KATA PENGANTAR

Segala puji bagi Allah SWT atas limpahan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan skripsi yang berjudul “Pengembangan Sistem Informasi Berbasis Web Penjurian Online Untuk Pengelolaan Data Kegiatan Pupuk Kaltim Inovation Award Pada Departement Inovasi Dan Pengembangan Manajemen Pt. Pupuk Kalimantan Timur Timur”.

Untuk kesempatan ini penulis juga menyampaikan rasa terima kasih kepada pihak-pihak yang telah membantu penulis selama penyusunan skripsi, diantaranya:

1. Allah SWT yang telah memberi kemudahan dalam semua proses penulisan skripsi ini.
2. Kepada kedua orang tua penulis, yaitu Bapak Hartono dan Ibu Darul Uswiyati beserta keluarga besar yang selalu memberikan segala masukan, do’a, motivasi dan semangat yang tidak terputus.
3. Bapak Fajar Pradana, S.ST., M.Eng selaku dosen pembimbing I yang telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
4. Bapak Adam Hendra Brata, S.Kom., M.T., M.Sc selaku dosen pembimbing II yang juga telah meluangkan waktu untuk memberikan masukan, ilmu, serta saran yang bermanfaat dalam proses penyelesaian skripsi ini.
5. Bapak Wayan Firdaus Mahmudy, S.Si, M.T, Ph.D, Bapak Ir. Heru Nurwasito, M.Kom, Bapak Drs. Mardji, M.T, dan Bapak Edy Santoso, S.Si, M.Kom selaku Dekan, Wakil Dekan 1, Wakil Dekan 2, dan Wakil Dekan 3 Fakultas Ilmu Komputer, Universitas Brawijaya.
6. Bapak Tri Astoto Kurniawan, S.T, M.T, Ph.D dan Bapak Agus Wahyu Widowo, S.T, M.Cs selaku Ketua Jurusan Teknik Informatika dan Kepala Program Studi Teknik Informatika Fakultas Ilmu Komputer, Universitas Brawijaya.
7. Seluruh Dosen Fakultas Ilmu Komputer, Universitas Brawijaya atas kesediaannya mengajarkan dan membagikan ilmu yang bermanfaat bagi penulis.
8. Karyawan PT. Pupuk Kaltim Departemen inbangmen bagian inovasi yang telah menjadi narasumber serta responden dalam pengujian yang dilakukan pada penelitian ini.
9. Semua pihak yang tidak semuanya bisa dituliskan disini yang terlibat secara langsung maupun tidak langsung dalam proses pengerjaan skripsi maupun sebagai pemberi semangat dan motivasi.

Penulis menyadari masih banyak kekurangan dalam penyusunan skripsi ini baik dalam teknik penyajian materi maupun pembahasan. Demi kesempurnaan penelitian skripsi ini, saran dan kritik yang sifatnya membangun sangat penulis harapkan. Semoga karya tulis ini bermanfaat dan dapat memberikan sumbangan yang berarti bagi pihak yang membutuhkan.

Malang, 23 Maret 2018

Penulis
farahshofiatul@gmail.com



ABSTRAK

PT. Pupuk Kalimantan Timur setiap tahunnya mengadakan kegiatan Pupuk Kaltim *Innovation Award* (PIA) yang menjadi wadah untuk partisipasi karyawan agar memunculkan inisiatif dalam melakukan inovasi pada semua lini aktifitas perusahaan untuk peningkatan mutu. Berdasarkan wawancara dengan pengelola PIA dapat disimpulkan bahwa sampai saat ini seluruh kegiatan PIA dilakukan secara manual, diantaranya adalah proses pengecekan originalitas makalah yang dilakukan oleh tim telaah sehingga mengakibatkan pelaksanaan kurang efisien karena banyaknya makalah yang dikaji, pemilihan juri yang belum memiliki aturan prioritas pemilihan sebagai juri tetap sehingga juri dapat berganti secara mendadak sehingga mengakibatkan aspek konsistensi pada kegiatan penilaian akan rendah, serta proses penilaian sehingga seorang juri harus selalu menyesuaikan *form* yang digunakan saat melakukan penilaian dengan jenis gugus yang sedang dinilai karena setiap jenis gugus memiliki form penilaian yang berbeda. Untuk menangani permasalahan-permasalahan tersebut, solusi yang dapat diterapkan adalah dengan mengembangkan sistem penjurian online berbasis web sesuai dengan fitur cek plagiasi internal, rekomendasi juri, serta penilaian online. Penelitian ini dimulai dengan menganalisa kebutuhan sistem, kemudian melakukan tahap perancangan dan implementasi. Pada penelitian ini menggunakan algoritma *cosine similarity* untuk penerapan fitur cek plagiasi internal, dan algoritma *weighted product* untuk penerapan fitur rekomendasi juri. Selanjutnya dilakukan tahap pengujian sistem dengan menggunakan pengujian unit, pengujian validasi, serta pengujian usabilitas. Hasil dari pengujian validasi menunjukkan bahwa sistem dapat bekerja dengan baik sesuai dengan kebutuhan, dan untuk hasil pengujian usabilitas menunjukkan hasil perhitungan skor 95,2% sehingga dapat disimpulkan bahwa peran sistem dalam mempercepat dan memudahkan kegiatan penjurian ini sudah terpenuhi.

Kata kunci: *cosine similarity*, sistem cek plagiasi, sistem rekomendasi, pengembangan sistem, *weighted product*.

ABSTRACT

PT. Pupuk Kalimantan Timur annually held Pupuk Kaltim Inovation Award (PIA) which is a forum for employee to participate in order to create inovation in all lines of corporate activitis for quality improvement. Based on interviews with PIA managers can be concluded that until now all PIA activities are done manually, such as checking the originality of the paper by tim telaah that can be less efficient implementation due to the number of papers, has no priority selection rule for determining permanent judges so the judges can change unexpectedly that can be less consistency in the assessment activities, and the assessment process that judges have to adjust the form when doing the assesment with the type of clster that is assessed because each type has different assessment form. To cover the issues is to develop a website with internal plagiarism check feature, judges rekomendations feature, and online assessment feature. This research begins by analyzing system requirements, design then implementation. In this research using cosine similarity algorithm for the application of internal plagiarism check feature, and weighted product algorithm for application of judges recommendation feature. The testing is done by using unit testing, validation testing, and usability testing. The result of the validation test shows that the system can work well in accordance with the needs, and for the results of usability tests show the result of the score of 95.2% so can be concluded that the role of the system in accelerating and facilitate the activity of this juries have been met.

Keywords: *cosine similarity, plagiarism checker system, rekomendation system, software enggineering, weighted product.*

DAFTAR ISI

| | |
|--|-------|
| PENGESAHAN | i |
| PERNYATAAN ORISINALITAS | ii |
| KATA PENGANTAR..... | iii |
| ABSTRAK..... | v |
| ABSTRACT | vi |
| DAFTAR ISI | vii |
| DAFTAR GAMBAR..... | xii |
| DAFTAR TABEL..... | xiv |
| DAFTAR KODE PROGRAM | xviii |
| DAFTAR LAMPIRAN | xix |
| BAB 1 PENDAHULUAN..... | 16 |
| 1.1 Latar belakang..... | 16 |
| 1.2 Rumusan masalah..... | 18 |
| 1.3 Tujuan | 18 |
| 1.4 Manfaat..... | 18 |
| 1.5 Batasan masalah | 19 |
| 1.6 Sistematika pembahasan..... | 19 |
| BAB 2 TINJAUAN PUSTAKA..... | 21 |
| 2.1 Kajian Pustaka | 21 |
| 2.2 Pupuk Kaltim <i>Inovation Award</i> | 22 |
| 2.2.1 Tahapan Penyusunan Inovasi..... | 22 |
| 2.2.2 Jenis-Jenis Gugus Inovasi | 24 |
| 2.2.3 Alur PIA..... | 25 |
| 2.2.4 Metode Penjurian | 26 |
| 2.3 Algoritme Pendukung Sistem | 27 |
| 2.3.1 <i>Preprocessing Text</i> | 27 |
| 2.3.2 Algoritme <i>Cosine Similarity</i> | 29 |
| 2.3.3 Algoritme <i>Weighted Product</i> | 29 |
| 2.4 Rekayasa Perangkat Lunak | 31 |
| 2.5 PHP | 31 |

| | |
|--|----|
| 2.5.1 Pengertian PHP | 31 |
| 2.5.2 Kelebihan PHP | 31 |
| 2.6 Framework CI (<i>CodeIgniter</i>) | 31 |
| 2.7 Software Development Life Cycle (SDLC) | 32 |
| 2.7.1 Waterfall Model | 32 |
| 2.8 Unified Modelling Language (UML) | 33 |
| 2.9 Pengujian | 34 |
| 2.9.1 Black Box Testing | 34 |
| 2.9.2 White Box Testing | 34 |
| 2.9.3 Usability Testing | 35 |
| BAB 3 METODOLOGI PENELITIAN | 37 |
| 3.1 Studi Literatur | 37 |
| 3.2 Pengumpulan Data | 38 |
| 3.3 Analisis Kebutuhan | 38 |
| 3.4 Perancangan Sistem | 38 |
| 3.5 Implementasi Sistem | 38 |
| 3.6 Pengujian | 39 |
| 3.7 Kesimpulan dan Saran | 39 |
| BAB 4 Analisis kebutuhan | 40 |
| 4.1 Identifikasi Actor | 40 |
| 4.2 Daftar Kebutuhan Sistem | 41 |
| 4.2.1 Daftar Kebutuhan Fungsional | 41 |
| 4.2.2 Daftar Kebutuhan Non Fungsional | 46 |
| 4.3 Pemodelan Kebutuhan | 46 |
| 4.3.1 Pemodelan <i>Use Case Diagram</i> | 46 |
| 4.4 <i>Use Case Scenario</i> | 47 |
| 4.4.1 Login | 47 |
| 4.4.2 Menilai Makalah | 47 |
| 4.4.3 Melihat Data Makalah <i>Stream</i> | 48 |
| 4.4.4 Melihat Hasil Penilaian Juri | 48 |
| 4.4.5 Melihat Nilai Akhir | 48 |

| | |
|---|--|
| 4.4.6 Mengunduh Makalah | 49 |
| 4.4.7 Logout | 49 |
| 4.4.8 Cek Plagiasi | 49 |
| 4.4.9 Melihat Makalah | 50 |
| 4.4.10 Menentukan Kelolosan Makalah | 50 |
| 4.4.11 Rekomendasi Juri | 51 |
| 4.4.12 Melihat Data Calon Juri | 51 |
| 4.4.13 Memilih Juri Tetap | 51 |
| 4.4.14 Menentukan <i>Stream</i> Juri | 52 |
| 4.4.15 Melihat Juri Tetap | 52 |
| 4.4.16 Menghapus Juri | 52 |
| 4.4.17 Menentukan <i>Stream</i> Makalah | 53 |
| 4.4.18 Melihat Data Makalah Terdaftar | 53 |
| 4.4.19 Menentukan Tim Telaah Makalah | 53 |
| 4.4.20 Melihat Data Makalah Lolos | 54 |
| 4.4.21 Melihat Data Tim Telaah | 54 |
| 4.4.22 Menambah Data Tim Telaah | 54 |
| 4.4.23 Menghapus Data Tim Telaah | 55 |
| 4.4.24 Melihat Data Pelatihan Calon Juri | 55 |
| 4.4.25 Menambah Data Pelatihan Calon Juri | 56 |
| BAB 5 PERANCANGAN DAN IMPLEMENTASI | 57 |
| 5.1 Perancangan | 57 |
| 5.1.1 Perancangan Arsitektur Sistem | 57 |
| 5.1.2 <i>Sequence Diagram</i> | 58 |
| 5.1.3 <i>Class Diagram</i> SIPO | 61 |
| 5.1.4 Perancangan Data | 61 |
| 5.1.5 Perancangan Algoritma | 62 |
| 5.1.6 Perancangan Antarmuka | 64 |
| 5.2 Implementasi | 78 |
| 5.2.1 Spesifikasi Pengembangan Sistem | 78 |
| 5.2.2 Implementasi Basis data | Error! Bookmark not defined. 79 |
| 5.2.3 Implementasi Kode Program | 80 |

| | |
|---|-----|
| 5.2.4 Implementasi Antarmuka | 84 |
| BAB 6 PENGUJIAN | 90 |
| 6.1 Pengujian Unit..... | 90 |
| 6.1.2 Pengujian Unit kelas timTelaah Operasi cekPlagiasi() | 90 |
| 6.1.2 Pengujian Unit Kelas admin Operasi rekomendasiJuri()..... | 92 |
| 6.1.3 Pengujian Unit Kelas juri Operasi menilai() | 97 |
| 6.2 Pengujian Integrasi | 101 |
| 6.3 Pengujian Validasi | 104 |
| 6.3.1 Pengujian Validasi Login..... | 105 |
| 6.3.2 Pengujian Validasi Menilai makalah | 107 |
| 6.3.3 Pengujian Validasi Melihat data makalah <i>stream</i> | 108 |
| 6.3.4 Pengujian Validasi melihat hasil penilaian juri..... | 108 |
| 6.3.5 Pengujian Validasi Melihat hasil akhir | 108 |
| 6.3.6 Pengujian Validasi Mengunduh makalah..... | 109 |
| 6.3.7 Pengujian Validasi Cek plagiasi | 109 |
| 6.3.8 Pengujian Validasi Melihat makalah | 110 |
| 6.3.9 Pengujian Validasi Menentukan kelolosan makalah | 110 |
| 6.3.10 Pengujian Validasi Rekomendasi Juri..... | 111 |
| 6.3.11 Pengujian Validasi Melihat data calon juri..... | 111 |
| 6.3.12 Pengujian Validasi Melihat juri tetap..... | 111 |
| 6.3.13 Pengujian Validasi Menentukan <i>stream</i> juri..... | 112 |
| 6.3.14 Pengujian Validasi Memilih juri tetap | 112 |
| 6.3.15 Pengujian Validasi menghapus juri tetap | 112 |
| 6.3.16 Pengujian Validasi menentukan <i>stream</i> makalah..... | 113 |
| 6.3.17 Pengujian Validasi Melihat data makalah terdaftar | 113 |
| 6.3.18 Pengujian Validasi Menentukan tim telaah makalah | 113 |
| 6.3.19 Pengujian Validasi Melihat data makalah lolos | 114 |
| 6.3.20 Pengujian Validasi Melihat data tim telaah | 114 |
| 6.3.21 Pengujian Validasi Menambah data tim telaah | 114 |
| 6.3.22 Pengujian Validasi menghapus data tim telaah..... | 116 |
| 6.3.23 Pengujian Validasi Melihat data pelatihan calon juri | 117 |
| 6.3.24 Pengujian Validasi Menambah pelatihan calon juri | 117 |

| | |
|--|-----|
| 6.3.25 Pengujian Validasi Logout | 119 |
| 6.4 Pengujian Usabilitas..... | 120 |
| 6.4.1 Hasil Pengujian Usabilitas | 120 |
| 6.5 Analisis Hasil Pengujian..... | 121 |
| 6.5.1 Pengujian Unit..... | 121 |
| 6.5.2 Pengujian Integrasi..... | 122 |
| 6.5.3 Pengujian Validasi | 122 |
| 6.5.4 Pengujian Usabilitas | 122 |
| BAB 7 penutup | 123 |
| 7.1 Kesimpulan..... | 123 |
| 7.2 Saran | 123 |
| DAFTAR PUSTAKA..... | 124 |
| DAFTAR LAMPIRAN | 126 |



DAFTAR GAMBAR

| | |
|--|----|
| Gambar 2.1 Daur Hidup Penyusunan Inovasi. | 23 |
| Gambar 2.2 <i>Scope</i> Masing - Masing Gugus..... | 25 |
| Gambar 2.3 Alur PIA..... | 25 |
| Gambar 2.4 Tahapan SDLC..... | 32 |
| Gambar 2.5 <i>Waterfall Model</i> | 32 |
| Gambar 3.1 Tahapan Penelitian..... | 37 |
| Gambar 4.1 <i>Use Case Diagram</i> SIPO..... | 46 |
| Gambar 5.1 Diagram Arititektur MVC..... | 57 |
| Gambar 5.2 <i>Sequence diagram</i> cek plagiasi. | 58 |
| Gambar 5.3 <i>Sequence diagram</i> Rekomendasi juri..... | 59 |
| Gambar 5.4 <i>Sequence diagram</i> Menilai Makalah..... | 60 |
| Gambar 5.5 <i>Class diagram</i> SIPO..... | 61 |
| Gambar 5.6 <i>Entity Relationship Diagram</i> SIPO. | 62 |
| Gambar 5.8 Perancangan Antarmuka Halaman Login..... | 65 |
| Gambar 5.9 Perancangan Antarmuka Halaman Home Tim Telaah | 66 |
| Gambar 5.10 Perancangan Antarmuka Halaman Cek Plagiasi | 67 |
| Gambar 5.11 Perancangan Antarmuka Halaman Hasil Cek Plagiasi..... | 68 |
| Gambar 5.12 Perancangan Antarmuka Halaman Home Admin | 69 |
| Gambar 5.13 Perancangan Antarmuka Halaman Daftar Tim Telaah..... | 70 |
| Gambar 5.14 Perancangan Antarmuka Halaman Tambah Tim Telaah..... | 72 |
| Gambar 5.15 Perancangan Antarmuka Halaman Data Calon Juri | 73 |
| Gambar 5.16 Perancangan Antarmuka Halaman Home Juri | 75 |
| Gambar 5.17 Perancangan Antarmuka Halaman <i>Form</i> Penilaian Juri..... | 76 |
| Gambar 5.18 Perancangan Antarmuka Halaman Hasil Akhir Penilaian | 77 |
| Gambar 5.19 <i>Physical Data Model</i> basis data <i>requirement repository</i> | 79 |
| Gambar 5.21 Implementasi Antarmuka Halaman Login | 84 |
| Gambar 5.22 Implementasi Halaman Home Tim Telaah..... | 84 |
| Gambar 5.23 Implementasi Antarmuka Halaman Cek Plagiasi | 85 |
| Gambar 5.24 Implementasi Antarmuka Halaman Hasil Cek Plagiasi..... | 86 |
| Gambar 5.25 Implementasi Antarmuka Halaman Home Admin | 86 |
| Gambar 5.26 Implementasi Antarmuka Halaman Daftar Tim Telaah | 87 |

| | |
|--|-----|
| Gambar 5.27 Implementasi Antarmuka Halaman Tambah Tim Telaah | 87 |
| Gambar 5.28 Implementasi Antarmuka Halaman Data Calon Juri | 88 |
| Gambar 5.29 Implementasi Antarmuka Halaman Home Juri | 88 |
| Gambar 5.30 Implementasi Antarmuka Halaman <i>Form</i> Penilaian | 89 |
| Gambar 5.31 Implementasi Antarmuka Halaman Hasil Akhir Penilaian | 89 |
| Gambar 6.1 Tampilan hasil Pengujian integrasi method cekPlagiasi() dengan method cosine(). | 103 |
| Gambar 6.2 Tampilan hasil Pengujian integrasi method rekomendasiJuri() dengan method countVi(calonJuri)..... | 104 |



DAFTAR TABEL

| | |
|---|----|
| Tabel 2.1 Kategori Perolehan | 27 |
| Tabel 2.2 Keterangan Bobot Untuk Kriteria | 30 |
| Tabel 2.3 Dimensi Kemudahan Penggunaan (<i>Usability</i>) | 35 |
| Tabel 2.4 Dimensi Kualitas Interaksi (<i>Information quality</i>)..... | 36 |
| Tabel 2.5 Dimensi Kualitas Interaksi (<i>Interction quality</i>) | 36 |
| Tabel 4.1 Identifikasi Actor | 40 |
| Tabel 4.2 Spesifikasi Kebutuhan Juri | 41 |
| Tabel 4.3 Spesifikasi Kebutuhan Tim Telaah | 42 |
| Tabel 4.4 Spesifikasi Kebutuhan Administrator | 43 |
| Tabel 4.5 Kebutuhan Non Fungsional Sistem | 46 |
| Tabel 4.6 <i>Use Case Scenario</i> Login | 47 |
| Tabel 4.7 <i>Use Case Scenario</i> Menilai Makalah..... | 47 |
| Tabel 4.8 <i>Use Case Scenario</i> Melihat Data Makalah <i>Stream</i> | 48 |
| Tabel 4.9 <i>Use Case Scenario</i> Melihat Hasil Penilaian Juri | 48 |
| Tabel 4.10 <i>Use Case Scenario</i> Melihat Nilai Akhir..... | 48 |
| Tabel 4.11 <i>Use Case Scenario</i> Mengunduh Makalah | 49 |
| Tabel 4.12 <i>Use Case Scenario</i> Logout..... | 49 |
| Tabel 4.13 <i>Use Case Scenario</i> Cek Plagiasi..... | 49 |
| Tabel 4.14 <i>Use Case Scenario</i> Melihat Makalah | 50 |
| Tabel 4.15 <i>Use Case Scenario</i> Menentukan Kelolosan Makalah | 50 |
| Tabel 4.16 <i>Use Case Scenario</i> Rekomendasi Juri | 51 |
| Tabel 4.17 <i>Use Case Scenario</i> melihat Data Calon Juri | 51 |
| Tabel 4.18 <i>Use Case Scenario</i> Memilih Juri..... | 51 |
| Tabel 4.19 <i>Use Case Scenario</i> Menentukan <i>Stream</i> Juri..... | 52 |
| Tabel 4.20 <i>Use Case Scenario</i> Melihat Juri Tetap | 52 |
| Tabel 4.21 <i>Use Case Scenario</i> menghapus Juri | 52 |
| Tabel 4.22 <i>Use Case Scenario</i> Menentukan <i>Stream</i> Makalah | 53 |
| Tabel 4.23 <i>Use Case Scenario</i> Melihat Makalah Terdaftar | 53 |
| Tabel 4.24 <i>Use Case Scenario</i> Menentukan Tim Telaah Makalah | 53 |
| Tabel 4.25 <i>Use Case Scenario</i> Melihat Makalah Lolos | 54 |
| Tabel 4.26 <i>Use Case Scenario</i> Melihat Data Tim Telaah | 54 |

| | |
|---|-----|
| Tabel 4.27 <i>Use Case Scenario</i> Menambah Data Tim Telaah | 54 |
| Tabel 4.28 <i>Use Case Scenario</i> Menambah Data Tim Telaah | 55 |
| Tabel 4.29 <i>Use Case Scenario</i> Melihat Data Pelatihan Calon Juri | 55 |
| Tabel 4.30 <i>Use Case Scenario</i> Menambah Data Pelatihan Calon Juri | 56 |
| Tabel 5.1 <i>Pseudocode</i> method CekPlagiasi pada kelas timtelaah | 62 |
| Tabel 5.2 <i>Pseudocode</i> method rekomendasiJuri pada kelas admin | 63 |
| Tabel 5.3 <i>Pseudocode</i> method penilaian pada kelas juri | 64 |
| Tabel 5.5 penjelasan Perancangan Antarmuka Halaman Login | 65 |
| Tabel 5.6 Penjelasan Perancangan Antarmuka Halaman Home Tim Telaah | 66 |
| Tabel 5.7 Penjelasan Perancangan Antarmuka Halaman Cek Plagiasi | 67 |
| Tabel 5.8 Penjelasan Perancangan Antarmuka Halaman Hasil Cek Plaiasi | 68 |
| Tabel 5.9 Penjelasan Perancangan Antarmuka Halaman Home Admin | 69 |
| Tabel 5.10 Penjelasan Perancangan Antarmuka Halaman Daftar Tim Telaah | 71 |
| Tabel 5.11 Penjelasan Perancangan Antarmuka Halaman Tambah Tim Telaah .. | 72 |
| Tabel 5.12 Penjelasan Perancangan Antarmuka Halaman Data Calon Juri | 73 |
| Tabel 5.13 Penjelasan Perancangan Antarmuka Halaman Home Juri | 75 |
| Tabel 5.14 Penjelasan Perancangan Antarmuka Halaman <i>Form</i> Penilaian Juri ... | 76 |
| Tabel 5.15 Penjelasa Perancanan Antarmuka Halaman Hasil Akhir Penilaian | 77 |
| Tabel 5.16 Spesifikasi perangkat keras | 78 |
| Tabel 5.17 Spesifikasi peragkat lunak | 79 |
| Tabel 5.18 Penjelasan kode program fungsi cekPlagiasi pada kelas timTelaah | 80 |
| Tabel 5.19 penjelasan kode program fungsi rekomendasiJuri pada kelas admin | 82 |
| Tabel 5.20 Penjelasan kode program fungsi menilai pada kelas juri | 83 |
| Tabel 6.1 Hasil Pengujian Unit Kelas timTelaah Operasi cekPlagiasi() | 92 |
| Tabel 6.2 Hasil Pengujian Unit Kelas Admin Operasi rekomendasiJuri() | 95 |
| Tabel 6.3 Hasil Pengujian Unit Kelas juri Operasi menilai() | 99 |
| Tabel 6.4 Interaksi method cekPlagiasi() dengan method cosine() | 101 |
| Tabel 6.5 Hasil Pengujian Integrasi Method cekPlagiasi() pada kelas timTelaah dan Method cosine() pada kelas similarity | 102 |
| Tabel 6.6 Interaki method rekomendasiJuri() dengan method countVi() | 103 |
| Tabel 6.7 Hasil Pengujian Integrasi Method rekomendasiJuri() pada kelas admin dan Method countVi() pada kelas weightedProduct | 104 |
| Tabel 6.8 Kasus uji berhasil melakukan login | 105 |

| | |
|---|-----|
| Tabel 6.11 kasus uji gagal melakukan login dengan tidak mengisi kolom password | 105 |
| Tabel 6.12 Kasus uji gagal melakukan login dengan tidak mengisi kolom NPK.. | 105 |
| Tabel 6.13 Kasus uji gagal melakukan login dengan masukan yang tidak sesuai pada kolom NPK | 106 |
| Tabel 6.14 Kasus uji gagal melakukan login dengan masukan yang tidak sesuai pada kolom password | 106 |
| Tabel 6.15 Kasus uji berhasil melakukan penilaian..... | 107 |
| Tabel 6.16 Kasus uji gagal melakukan penilaian | 107 |
| Tabel 6.17 Kasus uji berhasil melihat data makalah <i>stream</i> | 108 |
| Tabel 6.18 Kasus uji berhasil melihat penilaian juri..... | 108 |
| Tabel 6.19 Kasus uji berhasil melihat nilai akhir | 108 |
| Tabel 6.20 Kasus uji berhasil mengunduh makalah | 109 |
| Tabel 6.21 Kasus uji berhasil melakukan cek plagiasi..... | 109 |
| Tabel 6.22 Kasus uji gagal melakukan cek plagiasi | 109 |
| Tabel 6.23 Kasus uji berhasil makalah | 110 |
| Tabel 6.24 Kasus uji berhasil menentukan kelolosan makalah | 110 |
| Tabel 6.25 Kasus uji berhasil memberi rekomendasi juri | 111 |
| Tabel 6.26 Kasus uji berhasil melihat data calon juri..... | 111 |
| Tabel 6.27 Kasus uji berhasil melihat juri tetap..... | 111 |
| Tabel 6.28 Kasus uji berhasil menentukan <i>stream</i> juri..... | 112 |
| Tabel 6.29 Kasus uji berhasil memilih juri tetap | 112 |
| Tabel 6.30 Kasus uji berhasil menghapus juri | 112 |
| Tabel 6.31 Kasus uji berhasil menentukan <i>stream</i> makalah | 113 |
| Tabel 6.32 Kasus uji berhasil melihat data makalah terdaftar | 113 |
| Tabel 6.33 Kasus uji berhasil menentukan tim telaah makalah | 113 |
| Tabel 6.34 Kasus uji berhasil melihat data makalah lolos | 114 |
| Tabel 6.35 Kasus uji berhasil melihat data tim telaah | 114 |
| Tabel 6.36 Kasus uji berhasil menambah data tim telaah | 114 |
| Tabel 6.37 Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom npk..... | 115 |
| Tabel 6.38 Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom password | 115 |

| | |
|---|-----|
| Tabel 6.39 Kasus uji gagal menambahkan data tim telaah dengan mengisi kolom npk dengan data yang sudah terdapat pada data tim telaah..... | 116 |
| Tabel 6.40 Kasus uji berhasil menghapus data tim telaah..... | 116 |
| Tabel 6.41 Kasus uji berhasil melihat data pelatihan calon juri | 117 |
| Tabel 6.42 Kasus uji berhasil menambah pelatihan calon juri..... | 117 |
| Tabel 6.43 Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom pelatihan | 117 |
| Tabel 6.44 Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom tahun pelatihan..... | 118 |
| Tabel 6.45 Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom tahun pelatihan..... | 119 |
| Tabel 6.46 Kasus uji berhasil logout..... | 119 |
| Tabel 6.47 Bobot <i>skala Likert</i> untuk setiap pertanyaan | 120 |
| Tabel 6.48 Rekap Hasil Pengisian Kuisisioner..... | 120 |



DAFTAR KODE PROGRAM

| | |
|---|----|
| Kode Program 5.1 Kode Program fungsi cekPlagiasi pada kelas timTelaah..... | 80 |
| Kode Program 5.2 Kode Program fungsi rekomendasiJuri pada kelas admin | 81 |
| Kode Program 5.3 Kode Program fungsi menilai pada kelas juri..... | 82 |



DAFTAR LAMPIRAN

| | |
|--|-----|
| Lampiran A Daftar Juklak Penilaian..... | 126 |
| Lampiran B Form Penilaian SS dan SSG..... | 131 |
| Lampiran C Form Penilaian FAME..... | 134 |
| Lampiran D Form Penilaian PKM dan GKM..... | 137 |
| Lampiran E Hasil Pengisian Kuisisioner | 140 |



BAB 1 PENDAHULUAN

1.1 Latar belakang

PT. Pupuk Kalimantan Timur adalah salah satu anak perusahaan dari PT. Pupuk Indonesia (Persero) yang lahir untuk memenuhi kebutuhan pupuk yang semakin meningkat seiring dengan tingginya perkembangan pertanian di Indonesia. PT. Pupuk Kalimantan Timur merupakan perusahaan penghasil Urea dan Amoniak terbesar di Indonesia, dengan kapasitas produksi mencapai 3,43 juta ton Urea dan 2,765 juta ton Amoniak, 350 ribu ton NPK, dan 45 ribu ton pupuk organik per tahun. Semakin meningkatnya kebutuhan urea, maka perusahaan pupuk di Indonesia, khususnya PT. Pupuk Kalimantan Timur akan selalu meningkatkan kualitas produknya dengan inovasi-inovasi baru yang dapat menjaga kualitas serta untuk mempertahankan mutu dalam persaingan industri Internasional. Di bawah oleh Departemen Inovasi dan Pengembangan Manajemen, Bagian Inovasi dan Pengembangan Mutu Terpadu (PMT), PT. Pupuk Kalimantan Timur setiap tahunnya mengadakan kegiatan Pupuk Kaltim *Innovation Award* (PIA) yang menjadi wadah untuk partisipasi karyawan agar memunculkan semangat dan inisiatif dalam melakukan inovasi, perbaikan, dan penyempurnaan dalam proses produksi, proses bisnis, serta kualitas produk dan jasa di semua lini aktifitas perusahaan sehingga dapat memberikan nilai tambah bagi pemangku kepentingan. Kegiatan PIA ini juga akan digunakan sebagai *sharing of knowledge* dengan perusahaan lain di ajang Konvensi Mutu Nasional maupun Internasional. Pada tahun 2017 terdapat gugus yang mendaftarkan inovasinya sejumlah 150 (Pratama, 2017). Pada kegiatan PIA terdapat alur yang digunakan dalam pelaksanaannya yaitu pendaftaran gugus, pengumpulan makalah, tim telaah (tim yang berisi para *expert* untuk menelaah makalah yang dikumpulkan), penjurian lapangan, presentasi, *docking* (rapat pleno para juri untuk menentukan pemenang).

Pada tahap-tahap PIA yang berlangsung, terdapat tahapan penilaian oleh tim telaah yang dilakukan untuk mengkaji semua makalah-makalah yang terkumpul. Salah satu kegiatan pada tahapan tim telaah ini adalah pengecekan plagiasi makalah untuk mengecek originalitas dan memastikan bahwa makalah bukan dari hasil plagiat. Plagiat adalah kegiatan pengambilan karangan, pendapat, dan sebagainya dari orang lain yang diakui dan menjadikannya seolah-olah karangan dan pendapat sendiri (Alamanda, et al., 2016). Pada pelaksanaannya kegiatan penilaian oleh tim telaah ini proses pengecekan plagiasi masih belum dilaksanakan secara efisien dikarenakan banyaknya makalah yang dikaji dan makalah-makalah yang sudah ada sebelumnya, sehingga tidak memungkinkan dilakukan cek plagiasi secara manual dan belum tersedianya alat untuk pengecekan plagiasi makalah dalam lingkup PIA (Pratama, 2017).

Kegiatan penjurian dilakukan dengan menilai makalah dan presentasi 60 besar gugus terbaik yang ditentukan oleh tim telaah. Kegiatan penjurian secara paralel dengan membagi 60 gugus terbaik menjadi 4 *stream* dengan masing-masing *stream* terdapat 3 juri, sehingga diperlukan 36 juri dengan 36 *form* penilaian. Pada pelaksanaannya pemilihan juri dilakukan dengan memilih karyawan PT. Pupuk Kaltim yang sudah mengikuti pelatihan juri pemula. Prioritas yang dipakai saat ini adalah calon juri siapa saja yang tidak sedang sibuk karena pada realitasnya calon juri banyak yang memiliki kesibukan masing-masing sehingga pergantian juri secara mendadak sangat mungkin terjadi dikarenakan kegiatan PIA dilakukan selama 2 hari, sehingga aspek konsistensi pada kegiatan penilaian akan rendah (Pratama, 2017).

Pada kegiatan penilaian makalah dan presentasi, penilaian masih dilakukan secara manual dengan *form* yang tercetak dalam bentuk *hardcopy* dan terdapat beberapa *form* yang harus diperhatikan oleh juri yaitu juklak kriteria penilaian serta *form* penilaian, sehingga seorang juri harus selalu memperhatikan juklak kriteria penilaian ketika memberikan nilai pada *form* penilaian. Selain itu *form* penilaian memiliki perbedaan kriteria penilaian untuk setiap jenis gugus, sehingga seorang juri harus selalu menyesuaikan *form* yang digunakan saat melakukan penilaian dengan jenis gugus yang sedang dinilai. Hal tersebut kurang efektif dalam pemafaatan waktu serta pada kenyataanya hal tersebut juga menyebabkan banyak juri yang tidak mengisi lengkap *form* penilaian, selain itu seorang juri juga harus menghitung nilai akhir masih secara manual dengan rumus yang sudah ditetapkan (Pratama, 2017).

Dari permasalahan-permasalahan tersebut, maka akan dikembangkan Sistem Informasi Penjurian Online dengan fitur-fitur untuk menyelesaikan masalah-masalah tersebut. Fitur-fitur tersebut adalah fitur untuk cek plagiasi lingkup PIA yang mana akan berfungsi untuk menilai tingkat plagiasi dari abstrak makalah yang dikumpulkan. Nilai tingkat plagiasi tersebut akan diberikan dalam bentuk persen. Fitur kedua adalah fitur rekomendasi calon juri berfungsi untuk memberikan rekomendasi calon juri kepada pengelola kegiatan sehingga pengelola kegiatan dapat memilih dan menugaskan calon juri sebagai juri tetap pada kegiatan PIA. Fitur ketiga adalah fitur penilaian berfungsi agar seorang juri agar dapat melakukan penilaian secara mudah dan cepat melalui *form* penilaian yang lebih interaktif dengan menampilkan secara langsung informasi-informasi yang ada ketika *pointer* ditujukan ke kriteria-kriteria penilaian maupun digit nilai yang terdapat pada *form* penilaian serta memberikan fungsi untuk menghitung secara otomatis nilai akhir dengan rumus yang sudah ditentukan.

Hasil yang diharapkan pada penelitian ini adalah menghasilkan sistem informasi yang lebih terintegrasi untuk menjalankan proses kegiatan Pupuk Kaltim *Inovation Award* dengan cara mempermudah dan mempercepat kinerja juri dalam melakukan penilaian, membantu dalam hal penilaian originalitas karya inovasi dengan menggunakan cek plagiasi pada sistem sehingga mempercepat proses pengkajian karya inovasi, membantu menetapkan juri secara konsisten sehingga tidak terjadi pergantian juri secara sembarang dan tiba-tiba.

Sistem informasi penjurian *online* ini merupakan pengembangan dari sistem yang sudah ada dan dikembangkan pada kegiatan PKL (Praktik Kerja Lapangan). Pada sistem sebelumnya telah dibangun sebuah sistem dengan fitur pendaftaran *online* yang dapat membantu mempermudah pekerjaan karyawan di Bagian Inovasi dan PMT untuk merekap data secara terintegrasi sehingga data pendaftaran tersimpan pada server perusahaan dan dapat diakses oleh seluruh pengelola tanpa harus melakukan *download* manual dan menyimpan di komputer pengelola, hal itu membantu pengelola dalam hal *maintenance* data pendaftar tiap tahunnya. Selain itu terdapat juga fitur untuk *reporting* data pendaftaran secara otomatis yang nantinya akan digunakan sebagai bahan evaluasi kinerja Departemen Inovasi dan Pengembangan Manajemen.

1.2 Rumusan masalah

1. Bagaimana hasil analisis kebutuhan, perancangan, dan implementasi Sistem Informasi Penjurian Online berbasis web untuk pengelolaan PIA?
2. Apakah Sistem Informasi Penjurian Online pengelolaan PIA dapat membantu tim telaah dalam menentukan persentase plagiasi suatu abstrak makalah?
3. Apakah Sistem Informasi Penjurian Online pengelolaan PIA dapat memberikan rekomendasi juri tetap?
4. Apakah Sistem Informasi Penjurian Online pengelola PIA dapat memudahkan dan mempercepat kegiatan penjurian pada kegiatan PIA?

1.3 Tujuan

Tujuan dari penelitian ini adalah untuk pengembangan sistem yang sudah ada agar lebih terintegrasi dan membantu mempercepat pengelolaan kegiatan PIA.

Terdapat pula tujuan khusus yang ingin dicapai pada penelitian ini, yaitu :

1. Mengetahui kebutuhan yang dibutuhkan sistem serta perancangan dan implementasi berdasarkan kebutuhan yang telah didefinisikan sebelumnya pada Sistem Informasi Penjurian Online berbasis web untuk pengelolaan PIA.
2. Mengetahui keberhasilan sistem dalam menentukan persentase plagiasi suatu abstrak makalah.
3. Mengetahui keberhasilan sistem dalam memberikan rekomendasi juri tetap berdasarkan pelatihan yang pernah diikuti oleh seorang calon juri.
4. Mengetahui kemudahan sistem dalam kegiatan penjurian pada kegiatan PIA.

1.4 Manfaat

Diharapkan dengan adanya perancangan Sistem Informasi Penjurian Online sebagai pengembangan Sistem Informasi pengelolaan Pupuk Kaltim *Inovatin Award* dengan pendekatan berorientasi objek ini :

1. Dapat membantu instansi dalam mempermudah dan mempercepat kegiatan PIA.

2. Dengan adanya sistem yang terintegrasi memudahkan pengelola kegiatan untuk menyajikan report serta memonitor setiap tahap dalam kegiatan dengan lebih mudah.

1.5 Batasan masalah

Ruang lingkup pembahasan hanya difokuskan pada pengembangan sistem yaitu pada tahap penjurian kegiatan Pupuk Kaltim *Inovation Award*. Data-data yang akan digunakan pada penelitian ini didapat dari Bagian Inovasi dan PMT Departement Inovasi dan Pengembangan Manajemen PT. Pupuk Kaliman Timur.

Adapun batasan masalah yang terdapat didalam sistem ini adalah sebagai berikut:

1. Sistem berbasis web dengan menggunakan bahasa pemrograman PHP, HTML, Javascript, dan menggunakan framework *CodeIgniter*.
2. Basis data sistem yang digunakan adalah MySQL.
3. Fitur cek plagiasi hanya memproses kata-kata berbahasa Indonesia.

1.6 Sistematika pembahasan

Dalam penulisan penelitian ini, untuk mencapai tujuan yang telah ditetapkan maka sistematika penulisannya adalah:

BAB I PENDAHULUAN

Dalam bab ini memuat Latar Belakang, Identifikasi Masalah, Rumusan Masalah, Tujuan Penelitian, Manfaat Penelitian, Batasan Penelitian, dan Sistematika Pembahasan.

BAB II TINJAUAN PUSTAKA

Dalam bab ini memuat kajian pustaka dan dasar teori yang digunakan sebagai refereni pendukung.

BAB III METODOLOGI PENELITIAN

Dalam bab ini membahas mengenai mtode, pendekatan, dan alur pekerjaan yang digunakan dalam penelitian.

BAB IV ANALISIS KEBUTUHAN

Dalam bab ini membahas tentang proses analisis kebutuhan Sistem Informasi Penjurian Online berbasis web berdasarkan proses bisnis yang terdapat dalam pengelolaan Pupuk Kaltim *Inovation Award*.

BAB V PERANCANGAN DAN IMPLEMENTASI

Dalam bab ini membAhas tentang perancangan Sistem Informasi Penjurian Online berbasis web berdasarkan proses bisnis yang terdapat dalam pengelolaan Pupuk Kaltim *Inovation Award* serta membahas tentang implementasi Sistem Informasi Penjurian Online berbasis web berdasarkan proses bisnis yang terdapat dalam pengelolaan Pupuk Kaltim *Inovation Award*.

1.6.1 nambah Data Pelatihan Calon Juri

Tabel 1.1 Use Case Scenario Menambah Data Pelatihan Calon Juri

| | |
|-------------------------|---|
| Actor | Admin. |
| Objective | Mengizinkan actor untuk menambah data pelatihan calon juri. |
| Pre-Condition | Actor telah berada pada halaman pelatihan calon juri. |
| Main Flow | <ol style="list-style-type: none">1. Actor memilih tombol tambah.2. Sistem menampilkan formulir dengan kolom nama pelatihan, tahun pelatihan, dan tingkat pelatihan.3. Actor mengisi kolom nama pelatihan, tahun pelatihan, dan tingkat pelatihan lalu memilih tombol tambah. |
| Alternative Flow | <ol style="list-style-type: none">1. Jika Actor memilih tombol tambah tetapi nama pelatihan, tahun pelatihan, dan tingkat pelatihan masih belum diisi maka akan menampilkan pesan "Harap isi bidang" pada kolom yang kosong. |
| Post-Condition | Actor berhasil menambahkan data pelatihan calon juri. |



BAB 2 TINJAUAN PUSTAKA

2.1 Kajian Pustaka

Penelitian yang relevan dan sudah pernah dilakukan sebelumnya terdapat pada Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer vol. 2, No. 4, April 2018, hlmn. 1611-1619 yang dilakukan oleh Osa Prosdya, Ismiarta Aknuranda, dan Fajar Pradana dengan judul “Pengembangan Sistem Reservasi Penginapan Untuk Youth Hostel Griya Brawijaya”. Penelitian ini membahas tentang pengembangan Sistem Informasi berbasis web dengan menggunakan *Object Oriented* serta *framework CodeIgniter* dalam melakukan analisis dan perancangan, serta implementasinya. Hasil dari penelitian tersebut adalah sebuah sistem yang dapat membantu pengelola Youth Hostel Griya Brawijaya untuk mengelola seluruh kegiatan transaksi pemesanan kamar seperti proses pengelolaan data pesanan, data tipe kamar, data kamar, data denda, data pembayaran, serta laporan rekapitulasi pesanan. Sehingga dengan sistem informasi tersebut proses bisnis pemesanan kamar pada Youth Hostel Griya Brawijaya dapat berjalan dengan optimal (Prisdya, et al., 2018).

Sedangkan penelitian terdahulu lainnya terdapat pada jurnal Informatika vol. 10 No.1, Juni 2014, hlmn 31-42 yang dilakukan oleh Radiant Victor Imbar, Adelia, Mewati Ayub, Aleander Rehatta dengan judul “Implementasi *Cosine Similarity* dan algoritma *Smith-Waterman* untuk Mendeteksi Kemiripan Teks”. Penelitian ini membahas tentang implementasi algoritme *Cosine Similarity* untuk memperhitungkan kemiripan teks berdasarkan kemunculan kata dan algoritme *Smith-Waterman* untuk memperhitungkan kemiripan teks berdasarkan urutan kata. Sebagai tambahan penelitian ini menggunakan algoritme *Nazief-Adriana* untuk mengubah kata-kata berbahasa Indonesia menjadi kata-kata dasar yang dapat di-*Steam (Stemming)* pada tahap *Preprocessing*. Pada proses *Preprocessing* dilakukan proses perubahan bentuk data yang terstruktur sembarang menjadi data yang terstruktur sesuai kebutuhan untuk proses dalam *Text Mining*. Tahap-tahap dalam *Preprocessing* adalah *tokenizing*, *Filtering*, dan *Stemming*. Hasil dari penelitian ini adalah aplikasi menampilkan nilai presentase kemiripan *text* dari dokumen satu dengan yang lainnya. Nilai presentase ditampilkan sebanyak 2 nilai. Nilai pertama menampilkan hasil algoritma *Cosine Similarity* untuk memperhitungkan kemiripan teks berdasarkan kemunculan kata dan nilai kedua menampilkan hasil algoritma *Smith-Waterman* untuk memperhitungkan kemiripan teks berdasarkan urutan kata (Imbar, et al., 2014).

Penelitian lain yang sudah pernah dilakukan tentang implementasi algoritme *weighted product* terdapat pada jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer vol.1, No.12, Desember 2017, hlm. 1518-1525 dengan judul “Sistem Rekomendasi Lowongan Pekerjaan Untuk *Fresh Graduate* Menggunakan Metode *Weighted Product* Berbasis Android” yang dilakukan oleh Dwi Astuti, Aryo Pinandito, dan Ratih Kartika Dewi. Pada penelitian ini membahas implementasi algoritme *weighted product* pada sistem yang digunakan untuk memilih lowongan

pekerjaan yang sesuai dengan kriteria pengguna. Keluaran yang dihasilkan oleh sistem ini berupa daftar lowongan pekerjaan yang telah diurutkan berdasarkan hasil perhitungan dengan metode *weighted product* dengan kriteria-kriteria yang dipilih sendiri oleh pengguna (Astuti, et al., 2017).

Berdasarkan kajian terhadap tiga penelitian sebelumnya maka didapat penerapan pengembangan Sistem Informasi berbasis web dengan menggunakan *Object Oriented* serta *framework CodeIgniter* dalam melakukan analisis dan perancangan, serta implementasinya. Pada sistem penjurian online ini juga akan menerapkan algoritma *Cosine Similarity* untuk memperhitungkan kemiripan teks berdasarkan kemunculan kata serta tahap-tahap *Preprocessing (tokenizing, Filtering, dan Stemming)* yang akan digunakan untuk mengecek tingkat plagiasi suatu abstrak dari makalah-makalah yang telah terkumpul pada kegiatan PIA.

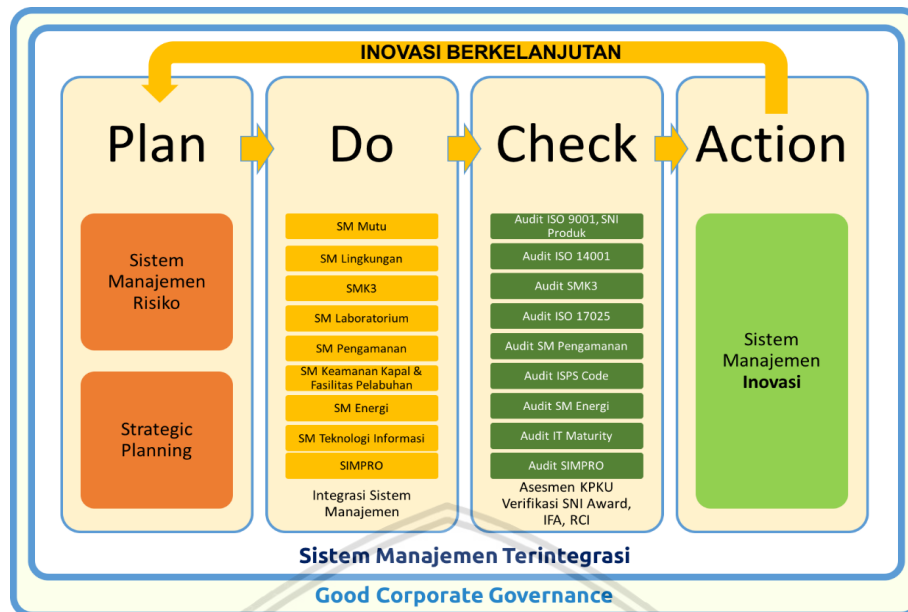
2.2 Pupuk Kaltim *Innovation Award*

Pupuk Kaltim *Innovation Award* (PIA) merupakan salah satu kegiatan rutin tahunan yang dilaksanakan di PT. Pupuk Kaltim dalam rangka melaksanakan program *continuous improvement* diharapkan dapat meningkatkan perbaikan kualitas yang pada akhirnya dapat memberikan nilai tambah bagi perusahaan. Tujuan diadakannya PIA antara lain yaitu:

1. Mewadahi proses *transfer of knowledge* karya mutu di lingkungan perusahaan dan sekaligus mempromosikan karya mutu terbaik.
2. Mendorong semangat dan inisiatif dalam melakukan inovasi, perbaikan, dan penyempurnaan dalam proses produksi, proses bisnis, serta kualitas produk dan jasa di semua lini aktifitas perusahaan tanpa mengabaikan unsur efektivitas dan efisiensi
3. Meningkatkan partisipasi seluruh karyawan dalam pemecahan masalah mutu
4. Mendayagunakan sumber daya manusia secara lebih baik guna meningkatkan kualitas dan produktivitas dan memberikan nilai tambah bagi *stakeholder*
5. Sebagai ajang pemilihan karya mutu terbaik perusahaan yang akan dipersiapkan untuk berpartisipasi dalam *sharing of knowledge* dengan perusahaan lain di ajang Konvensi Mutu Nasional maupun Internasional.

2.2.1 Tahapan Penyusunan Inovasi

Dalam rangka meningkatkan kinerja serta mengendalikan mutu perusahaan, perlu adanya inovasi baru untuk dapat memperbaiki kekurangan yang ada. Inovasi yang ada pada PT. Pupuk Kaltim memiliki daur hidup yang memutar dan berbasis *Plan Do Check Action* (PDCA). Ketika terdapat kinerja yang dinilai kurang baik, maka perlu ditingkatkan dengan metode PDCA. Kemudian ketika kinerja sudah meningkat, akan dilakukan standarisasi untuk menjaga agar kinerja yang sudah meningkat tersebut tidak menurun dengan menggunakan *Standarize Do Check Action* (SDCA).



Gambar 2.1 Daur Hidup Penyusunan Inovasi.

Pada Gambar 2.1 merupakan daur Hidup PDCA dijabarkan sebagai berikut:

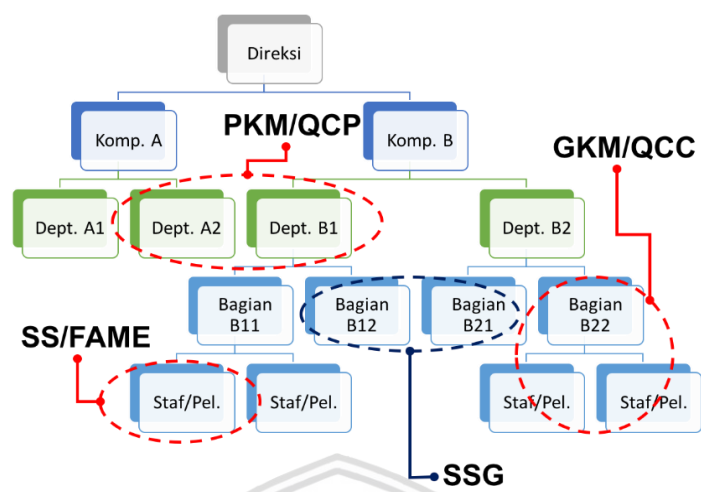
1. **Plan**
Pada tahap ini akan dibuat Rencana Kerja dan Anggaran Perusahaan (RKAP) tahunan yang diambil dari Rencana Jangka Panjang (RJP), RJP ini diambil dari *Corporate Plan*. Rencana kerja yang dibuat akan mengacu pada kondisi terakhir dari suatu perusahaan dimana sebelumnya telah dilakukan inovasi.
2. **Do**
Pada tahap ini akan dilakukan implementasi dari rencana kerja yang telah dibuat pada tahap sebelumnya dengan melakukan semua Sistem Manajemen untuk memutar roda operasi perusahaan.
3. **Check**
Setelah tahap implementasi selesai, selanjutnya akan dilakukan penilaian dan pemeriksaan apakah dalam proses implementasi berjalan dengan baik dan sesuai dengan rencana kerja yang telah disusun. Pemeriksaan ini dilakukan dengan cara audit setiap 2 atau 3 tahun serta adanya survei dari luar pada setiap tahunnya. Dengan adanya audit ini akan ditemukan kekurangan yang nantinya akan menjadi peluang perusahaan untuk melakukan perbaikan atau biasa disebut dengan *Opportunity For Improvement* (OFI). OFI inilah yang nantinya menjadi masukan atau menjadi dasar untuk dilakukannya Inovasi baru.
4. **Action**
Pada tahap ini akan dilakukan inovasi baru berdasarkan OFI untuk dapat memperbaiki peluang perbaikan yang ditemukan pada tahap *Check*. Dengan dibutuhkannya Inovasi baru ini maka PT. Pupuk Kaltim membuat suatu kegiatan guna menyalurkan inovasi-inovasi untuk meningkatkan kinerja perusahaan.

2.2.2 Jenis-Jenis Gugus Inovasi

Terdapat 5 jenis gugus yang dapat didaftarkan pada Pupuk Kaltim *Innovation Award*, antara lain:

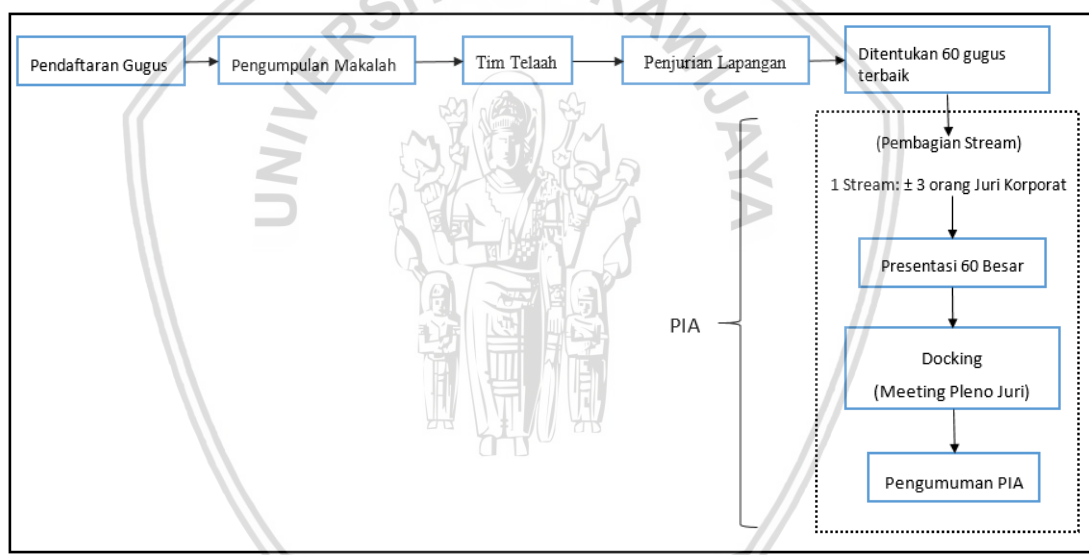
1. **Proyek Kendali Mutu (PKM)**
Pada gugus ini, inovasi yang dibuat berfokus pada lingkup lintas unit kerja atau lebih dari satu proses bisnis. Dalam pembuatan inovasi pada gugus PKM harus menguasai *tools-tools* dengan menggunakan metode Delapan Langkah Tujuh Alat atau biasa disebut DELTA yang berbasis PDCA. Peserta yang mendaftar pada gugus PKM biasanya beranggotakan 8 orang yang terdiri dari 1 Fasilitator, 1 Ketua, dan 6 anggota.
2. **Gugus Kendali Mutu (GKM)**
Pada gugus ini, inovasi yang dibuat hanya berfokus pada satu unit kerja saja. Dalam pembuatan inovasi pada gugus GKM harus menguasai *tools-tools* dengan menggunakan metode Delapan Langkah Tujuh Alat atau biasa disebut DELTA yang berbasis PDCA. Peserta yang mendaftar pada gugus GKM biasanya beranggotakan 6 orang yang terdiri dari 1 Fasilitator, 1 Ketua, dan 4 anggota.
3. **Sistem Saran Grup (SSG)**
Pada gugus ini inovasi yang dibuat berfokus pada lingkup lintas unit kerja, namun cakupan inovasi atau permasalahan yang ditangani tidak sebesar PKM. Dalam pembuatan inovasi pada gugus SSG harus menguasai *tools* dengan menggunakan metode Delapan Langkah Tujuh Alat atau biasa disebut DELTA berbasis PDCA. Peserta yang mendaftar pada gugus SSG biasanya beranggotakan 6 orang yang terdiri dari 1 Fasilitator, 1 Ketua, dan 4 anggota.
4. **Sistem Saran (SS)**
Pada gugus ini, inovasi yang dibuat hanya berfokus pada pekerjaan atau *job description* dari penggagas inovasi itu sendiri. Dalam pembuatan inovasi pada gugus SS harus menguasai *tools-tools* dengan menggunakan metode Delapan Langkah Tujuh Alat atau biasa disebut DELTA. Peserta yang mendaftar pada gugus SS biasanya beranggotakan maksimal 2 orang saja, ditambah adanya 1 orang fasilitator.
5. **Free Improvement Activity Method (FAME)**
Pada gugus ini inovasi yang dibuat hanya berfokus pekerjaan atau *job description* dari penggagas inovasi itu sendiri. Berbeda dengan keempat jenis gugus lainnya, dalam pembuatan inovasi pada gugus FAME tidak harus menguasai *tools* dengan menggunakan metode Delapan Langkah Tujuh Alat atau biasa disebut DELTA, melainkan hanya menggunakan daur PDCA saja. Peserta yang mendaftar pada gugus FAME biasanya beranggotakan maksimal 2 orang saja, ditambah adanya 1 orang fasilitator.

Dalam Gambar 2.2 menjelaskan *scope* masing-masing gugus:



Gambar 2.2 Scope Masing - Masing Gugus.

2.2.3 Alur PIA



Gambar 2.3 Alur PIA.

Untuk dapat mengikuti PIA, terdapat beberapa tahapan yang harus dijalani oleh para peserta seperti pada Gambar 2.3, antara lain:

1. Pendaftaran Gugus
Pada tahap ini, calon peserta harus mendaftarkan gugus yang akan diikuti dalam acara PIA dengan cara mengisi form pendaftaran secara *online*. Pada form pendaftaran, peserta akan diminta mengisi data seperti nama gugus, jenis gugus, identitas anggota, serta *upload* makalah sebagai salah satu syarat pendaftaran serta data-data lainnya.

2. Pengumpulan Makalah
Setelah melakukan pendaftaran secara online, peserta diwajibkan untuk mengumpulkan makalah yang sebelumnya telah di upload untuk nantinya akan dilakukan penilaian.
3. Tim Telaah
Pada tahap ini, makalah yang telah dikumpulkan akan di telaah oleh para expert yang telah ditunjuk oleh panitia penyelenggara PIA.
4. Penjurian Lapangan
Pada tahap ini akan dilakukan penjurian lapangan oleh Juri Direktorat dengan diadakannya wawancara kepada para peserta. Juri akan datang ke unit kerja masing-masing gugus yang mendaftar. Setelah itu akan dipilih 60 gugus terbaik dan dibagi menjadi 4 *Stream* (kelompok) yang nantinya akan berlaga di acara puncak Pupuk Kaltim *Innovation Award*.
5. Presentasi
Tahap presentasi merupakan tahap yang termasuk dalam acara puncak Pupuk Kaltim *Innovation Award*. Setelah 60 gugus terbagi menjadi 4 *Stream*, masing-masing gugus akan melakukan presentasi yang dinilai oleh Juri Korporat.
6. Docking
Pada tahap terakhir ini akan dilakukan rapat pleno oleh para juri-juri untuk menentukan peringkat secara konsensus dan mendapatkan pemenang dari acara PIA.

2.2.4 Metode Penjurian

Pada tahap presentasi pada alur PIA akan dilakukan presentasi sebanyak 60 besar gugus terpilih. Dari 60 gugus tersebut akan dibagi oleh pengelola menjadi 4 *stream* yang mana setiap stream akan dinilai oleh juri korporat. Maka dari itu setiap juri korporat akan bertanggung jawab untuk menilai sebanyak 12 gugus.

Metode yang digunakan untuk penjurian pada PIA ini adalah dengan memberikan bobot penilaian pada setiap tahapan penyusunan (*PLAN-DO-CHECK-ACTION*) serta presentasi dengan kriteria-kriteria yang dicantumkan pada juklak kriteria yang sudah ditetapkan dan tercantum pada lampiran A. Dalam Lampiran A tersebut penilaian terbagi menjadi 4 kriteria yang nantinya akan digunakan sebagai acuan untuk penjurian dengan form yang sudah disediakan. Kriteria-kriteria tersebut yaitu:

1. Kurang (1-4,9)
2. Cukup (5-6,9)
3. Baik (7-8,9)
4. Excellent (9-10)

Dalam lampiran B, C, dan D ditunjukkan bahwa pada *form* penilaian seorang juri harus mengisi kriteria-kriteria penilaian pada setiap tahapan penyusunan serta presentasi dengan kriteria penilaian sesuai dengan juklak pada lampiran A.

Metode penilaian yang digunakan adalah dengan memberi bobot pada setiap kriteria tahapan penyusunan dan presentasi yang kemudian jika dijumlahkan akan menghasilkan 100. Kemudian seorang juri akan memberi nilai pada setiap kriteria dan nilai tersebut akan dikalikan dengan bobot yang sudah ditetapkan. Nilai maksimal nantinya akan mencapai 1000 untuk menentukan kategori perolehan untuk setiap gugus yang dinilai. Kategori penilaian tersebut ditunjukkan pada Tabel 2.1.

Tabel 2.1 Kategori Perolehan

| Kategori Perolehan | Range Total Nilai Akhir |
|--------------------|-------------------------|
| Gold | 851 ~ 1000 |
| Silver | 701 ~ 850 |
| Bronze | 600 ~ 700 |

2.3 Algoritme Pendukung Sistem

2.3.1 Preprocessing Text

Preprocessing text adalah tahap awal dalam melakukan proses *text mining*. *Text Mining* adalah sebuah metode yang dapat digunakan untuk menemukan informasi baru yang dapat diambil secara jelas atau mudah dipahami dari sebuah dokumen (Konchady, 2006). Proses yang dilakukan pada *preprocessing* adalah perubahan data terstruktur sembarang menjadi data terstruktur sesuai kebutuhan. Tahap *preprocessing* terdiri dari :

1. Tokenisasi

Tokenisasi adalah pemisahan komponen-komponen struktur dokumen. Jika struktur sebuah dokumen adalah dalam bentuk kalimat, maka akan dipisah menjadi komponen-komponen kata (Weiss, et al., 2005). Pada proses ini tanda pemisah kata adalah spasi dan tidak memiliki pengaruh terhadap pemrosesan teks.

2. *Stemming*

Stemming adalah proses merubah bentuk kata berimbuhan menjadi kata dasar (Asian, 2007). Pada proses ini akan menghilangkan setiap awalan serta akhiran kata hasil dari proses filterisasi menjadi kata dasar. Terdapat aturan-aturan untuk menghilangkan awalan serta akhiran tersebut karena terdapat banyak masalah dalam proses *stemming* pada kata bahasa Indonesia (Asian, 2007). Aturan-aturan tersebut antara lain:

1. Imbuhan pada bahasa Indonesia yang cukup kompleks, terdiri dari:

- a. Prefiks, yaitu imbuhan di depan kata pada bahasa Indonesia yang terdiri dari di-, ke-, se-, ber-, be-, me-, pe-, ter-, dan sebagainya. Contoh: ter-sesat.
- b. Suffixs, yaitu imbuhan di akhir kata pada bahasa Indonesia yang terdiri dari -lah, -kah, -pun, -ku, -mu, -nya, -i, -kan, -an, dan sebagainya. Contoh: hancur-lah.

- c. Konfiks, yaitu imbuhan pada awal dan akhir kata. Contoh: per-usaha-an.
 - d. Infiks, yaitu imbuhan di tengah kata. Contoh: k-em-ilau.
 - e. Imbuhan dari bahasa asing. Contoh: sosial-isasi.
 - f. Aturan perubahan prefiks, seperti me- menjadi meng-, mem-, men-, atau meny- dan pe- menjadi per-, pem-, pen-, peng-, atau peny-.
2. *Word-Sense Ambiguity* (Ambiguitas Rasa Kata), yaitu satu kata dapat memiliki dua makna (seperti misalnya homonim) dan berasal dari dua kata yang berbeda. Contoh:
 - a. Berikan → Ber-ikan
 - b. Berikan → Beri-kan
 3. *Overstemming*, yaitu proses *stemming* yang berlebihan. Contohnya pada kata 'Berikan'. Jika dilakukan *stemming* berdasarkan aturan poin 1, maka hasilnya kata 'Berikan' akan menjadi kata 'Ber-i-kan' sehingga menghasilkan kata dasar 'i' yang tidak menghasilkan makna apa-apa. Oleh karena itu, sebuah kamus atau daftar kata dasar dibutuhkan dalam algoritme *stemming* untuk mencegah adanya *overstemming*. Dengan menggunakan kamus ini, apabila dalam proses penghapusan imbuhan atau *stemming* terdapat kata yang sama dengan yang ada di dalam kamus atau daftar kata dasar, maka proses *stemming* dihentikan.
 4. *Understemming*, yaitu proses *stemming* yang tidak sempurna sehingga tidak menghasilkan kata dasar yang tepat. Contoh: 'mengecek' di-*stem* menjadi 'meng-ecek', padahal seharusnya menjadi 'menge-cek'. Kata 'ecek' juga terdapat pada kamus atau daftar kata dasar sehingga lolos pada proses *stemming*. Namun bukan kata 'ecek' yang dimaksud dari kata 'mengecek' sehingga perlu perbaikan pada persoalan ini.
 5. Pengguna bahasa Indonesia tidak konsisten dalam menentukan *stem* (secara manual). Setiap orang terkadang berbeda opini dalam menentukan *stem* pada suatu kata. Contoh: apakah kata 'adalah' merupakan kata berimbuhan dari kata dasar dari kata 'ada'?
 6. Akronim, yaitu kependekan yang berupa gabungan huruf atau suku kata atau bagian lain yang ditulis dan dilafalkan sebagai kata yang wajar. Akronim dapat memberi permasalahan dalam proses *stemming* karena tidak ada algoritme yang dapat mendeteksi akronim tersebut dan melewati proses *stemming* sehingga tidak menghasilkan makna kata yang diharapkan. Contoh: pemilu → pe-milu.
 7. *Proper Noun* atau nama benda, misal nama orang yang dapat mempunyai kemungkinan mendapatkan proses *stemming* karena tidak ada algoritme untuk mendeteksi nama benda. Contoh: Abdullah → Abdul-lah.

Dari permasalahan-permasalahan diatas, maka diperlukan algoritme *preprocessing text* untuk tokenisasi dan *stemmer* yang dapat menangani permasalahan-permasalahan diatas. Salah satu algoritme *preprocessing text* adalah Sastrawi *Stemmer* dan Sastrawi *Tokenizer*. Sastrawi *Stemmer* adalah sebuah

library stemmer yang menerapkan beberapa algoritme seperti algoritme Nazief Adriani, lalu ditingkatkan dengan algoritme *Confix Stripping*, lalu ditingkatkan lagi dengan algoritme *Enhance Confix Stripping*, lalu ditingkatkan lagi dengan *Modified Enhance Confix Stripping*. Dengan algoritme-algoritme tersebut maka stemmer tersebut sudah dapat menyelesaikan permasalahan-permasalahan diatas.

2.3.2 Algoritme Cosine Similarity

Metode *Cosine Similarity* adalah untuk menghitung similaritas pada dua dokumen. Penentuan kesesuaian dokumen dengan *query* dipandang sebagai pengukuran (*similarity measure*) antara *vector document* (D) dengan *vector query* (Q). Semakin sama suatu *vector document* dengan *vector query* maka dokumen dapat dipandang semakin sesuai dengan *query*. Rumus yang digunakan untuk menghitung cosine similarity ditunjukkan pada rumus 2.1 sebagai berikut:

$$f(X, d_j) = \frac{\sum_{i=1}^m x_i \cdot d_{ji}}{\sqrt{(\sum_{i=1}^m x_i)^2} \sqrt{(\sum_{i=1}^m x_i d_{ji})^2}} \quad (2.1)$$

Pada rumus 2.1 tersebut X adalah dokumen uji, d_j adalah dokumen penguji, x_i dan d_{ji} adalah nilai bobot yang diberikan pada setiap *term* pada dokumen. Kedekatan *query* dan dokumen diindikasikan dengan sudut yang dibentuk. Nilai *cosinus* yang cenderung besar mengindikasikan bahwa dokumen cenderung sesuai *query*. Dalam proses membandingkan dokumen yang sesuai dengan dokumen yang telah ada atau dokumen lainnya, maka digunakan perhitungan dengan rumus pada persamaan (2.1) untuk mengetahui angka similaritas dari dokumen tersebut.

2.3.3 Algoritme Weighted Product

Metode *Weighted Product* merupakan suatu metode yang digunakan untuk menentukan sebuah keputusan terbaik dari beberapa alternatif yang ada (Astuti, et al., 2017). Dalam penelitian ini algoritme *Weighted Product* digunakan untuk memberikan rekomendasi calon juri berdasarkan pelatihan-pelatihan yang pernah diikuti dan akan digunakan untuk membantu menentukan juri tetap. Sistem Rekomendasi adalah aplikasi penyedia saran yang berhubungan dengan berbagai proses dalam pengambilan keputusan (Astuti, et al., 2017). Metode ini dilakukan dengan cara mengalikan untuk menghubungkan rating kriteria, rating tiap kriteria harus dipangkatkan dahulu dengan bobot kriteria yang bersangkutan. Hal tersebut sama dengan proses normalisasi. Setelah mendapatkan data kriteria yang digunakan, preferensi alternatif dihitung menggunakan rumus Persamaan 2.2 (Nofriansyah, 2015):

$$S_i = \prod_{j=1}^n X_{ij}^{W_j} \quad (2.2)$$

Keterangan:

S : Preferensi alternatif atau vektor S

X : Nilai kriteria

W : Bobot kriteria atau sub kriteria

i : Alternatif ($i = 1, 2, 3, \dots, n$)

j : Kriteria

n : Banyaknya kriteria

W_j dapat dihitung dengan menggunakan rumus seperti pada persamaan 2.3

$$W_j = \frac{w_j}{\sum w_j} \quad (2.3)$$

Kemudian dilakukan proses perhitungan untuk menentukan ranking dari alternatif seperti pada persamaan 2.4

$$V_i = \frac{\prod_{j=1}^n x_{ij}^{w_j}}{\prod_{j=1}^n (x_j^*)^{w_j}} \quad (2.4)$$

V : Preferensi alternatif dianalogikan sebagai vektor V

X : Nilai Kriteria

W : Bobot kriteria atau sub kriteria

i : Alternatif

j : Kriteria

n : Banyaknya kriteria

$*$: Banyaknya kriteria yang telah dihitung pada vektor S

Pada Tabel 2.4 menjelaskan keterangan tiap kriteria yang akan dipakai pada penelitian ini.

Tabel 2.2 Keterangan Bobot Untuk Kriteria

| Kriteria | Bobot |
|-------------------------|-------|
| Pelatihan Internasional | 3 |
| Pelatihan Nasional | 2 |
| Pelatihan internal | 1 |

Pembobotan dari setiap kriteria pada Tabel 2.2 tersebut berdasarkan tingkat pelatihan yang telah diikuti oleh calon juri. Terdapat 3 tingkat pelatihan yang ditentukan sebagai kriteria yaitu tingkat pelatihan Internasional, tingkat pelatihan nasional, dan tingkat pelatihan internal. Bobot yang diberikan pada setiap kriteria berbeda-beda diantaranya untuk pelatihan internasional bernilai 3, untuk pelatihan nasional bernilai 2, dan untuk pelatihan internal bernilai 1. Pada pembobotan ini, bobot tingkat pelatihan diurutkan berdasarkan lingkup pelatihan dari lingkup yang luas hingga kecil. Pada pelatihan dengan lingkup yang lebih luas akan mendapatkan bobot yang lebih besar karena dengan lingkup pelatihan yang luas maka akan seorang calon juri menghasilkan pengalaman dan pengetahuan yang lebih dibandingkan dengan lingkup pelatihan yang lebih kecil, sehingga juri yang mengikuti pelatihan dengan tingkat pelatihan yang memiliki lingkup lebih besar akan lebih dipertimbangkan daripada pelatihan dengan tingkat pelatihan yang lainnya.

2.4 Rekayasa Perangkat Lunak

Rekayasa perangkat lunak sebagai disiplin ilmu yang mengadopsi pendekatan rekayasa seperti metodologi, proses, alat, metode organisasi, standar, metode manajemen, sistem jaminan kualitas, dan pengembangan perangkat lunak skala besar dengan nilai produktivitas yang tinggi, biaya rendah, kualitas terkontrol, dan pengukuran jadwal pengembangan (Hepyan, et al., 2012). Factor-factor umum yang melibatkan perangkat lunak antara lain (Pressman, 2009):

1. Sistem Perangkat lunak dibangun oleh tim bukan individu.
2. Dalam pengembangan sistem menggunakan prinsip teknik yang mencakup baik aspek teknis maupun non-teknis.

2.5 PHP

2.5.1 Pengertian PHP

PHP (*Hypertext Preprocessor*) adalah sebuah bahasa pemrograman digunakan sebagai *script server-side* dalam pengembangan *web* yang dituliskan pada dokumen HTML (*Hypertext Markup Language*). PHP memungkinkan web dapat dikembangkan dan dibangun secara dinamis sehingga web lebih mudah dan efisien dalam *maintenance* (Sidik, 2006).

2.5.2 Kelebihan PHP

PHP dapat melakukan hal-hal seperti mendapatkan data dari sebuah *form*, menerima *cookies*, dan menghasilkan web yang bersifat dinamis. Kelebihan PHP yang paling diandalkan adalah kemampuan PHP yang dapat memberi dukungan kepada banyak basis data. Hal tersebut membuat web dapat mengakses basis data dengan sangat mudah (Arshad, 2011).

2.6 Framework CI (CodeIgniter)

Framework CodeIgniter adalah sebuah *Framework* yang terbentuk dalam format PHP. Penggunaan *CodeIgniter* dapat mempercepat kinerja dalam proses pengembangan web karena sudah tersedianya kelas serta *modul-modul* yang dibutuhkan sehingga dapat digunakan langsung oleh pengembang web (Lukmanul, 2010). Dalam *framework CodeIgniter* alur program menggunakan pola MVC.

MVC merupakan sebuah metode pengembangan aplikasi yang memisahkan bagian data (*Model*), tampilan (*View*), dan proses (*Controller*). Penggunaan MVC ini bertujuan untuk mempermudah pengembang berikutnya untuk memahami kode program pada saat dilakukan pengembangan (Prabowo, 2015).

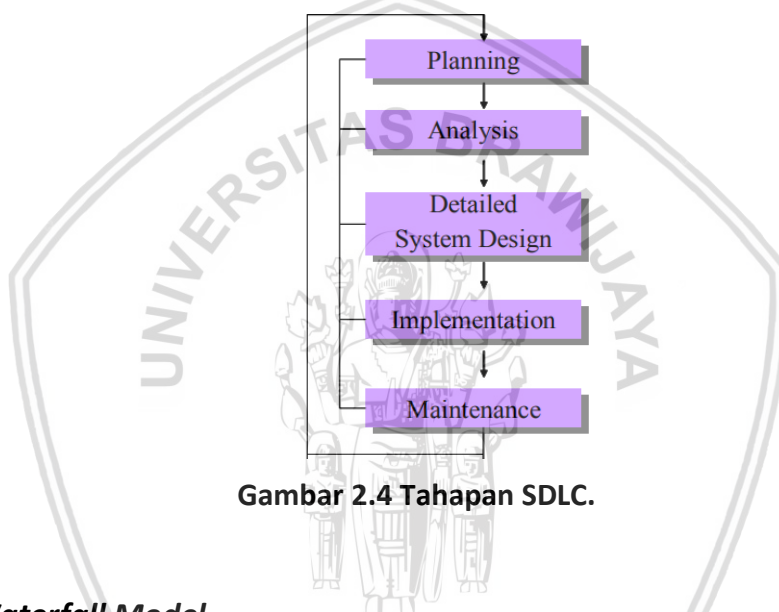
Penggunaan *framework CodeIgniter* untuk pengembangan sistem dalam penelitian ini karena pada *framework CodeIgniter* memiliki beberapa kelebihan antara lain (Noviyanto, 2010):

1. Merupakan *software* yang bersifat *open source* yang berada dibawah *Apache/BSD-style open source licence* sehingga dapat diperoleh secara gratis.

2. Memiliki ukuran yang kecil sehingga dapat berjalan secara ringan karena dalam penggunaannya bisa diatur agar sistem hanya me-load *library* yang dibutuhkan saja.
3. Menyediakan fungsi dan kelas untuk memudahkan dalam melakukan koneksi terhadap database.

2.7 Software Development Life Cycle (SDLC)

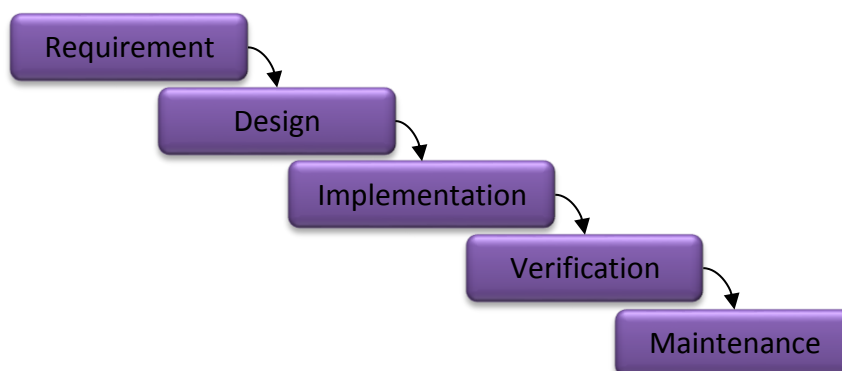
SDLC adalah suatu urutan dari beberapa proses atau kegiatan yan berurutan dalam pengembangan sistem. SDLC menggunakan prinsip siklus (*cycle*) karena pada kenyataanya suatu sistem akan selalu memiliki perubahan-perubahan seperti perkembangan teknologi, proses bisnis, dan lainnya (Dewanto, 2004). Pada Gambar 2.4 digambarkan alur dari tahapan-tahapan SDLC:



Gambar 2.4 Tahapan SDLC.

2.7.1 Waterfall Model

Model *waterfall* merupakan salah satu model pendekatan SDLC yang pertama kali terbentuk dan memiliki prinsip kerja secara linier dan berurutan dimulai dari analisis kebutuhan, desain, implementasi, pengujian, dan *maintanance* (Pressman, 2009).



Gambar 2.5 Waterfall Model.

Pada Gambar 2.5 digambarkan alur dari tahapan *waterfall Model* yang terdiri dari:

1. Analisis kebutuhan

Dilakukan dengan pelanggan secara intensif untuk dapat memahami kebutuhan dari pengguna dan untuk menentukan spesifikasi kebutuhan sistem sebagai langkah awal dalam pengembangan sistem.

2. Desain

Dilakukan dengan cara merepresentasikan kebutuhan-kebutuhan yang sudah didefinisikan menjadi sebuah desain pengembangan sistem agar dapat dipahami pada saat tahap implementasi. Proses tersebut antara lain meliputi perancangan struktur data, arsitektur sistem, dan perancangan antarmuka.

1. Implementasi

Dilakukan dengan cara membuat code program sesuai dengan desain yang sudah ada sehingga dapat memenuhi semua kebutuhan yang telah didefinisikan.

2. Pengujian

Dilakukan pada hasil implementasi sistem. Hal ini dilakukan untuk meminimalisir kesalahan-kesalahan dan memastikan bahwa hasil implementasi telah memenuhi semua kebutuhan yang telah didefinisikan.

3. *Maintenance*

Dilakukan pada tahap akhir setelah sistem telah disetujui oleh pelanggan. Pada tahap ini dilakukan proses instalasi serta pemeliharaan terhadap sistem yang tidak menutup kemungkinan sistem tersebut mengalami perubahan yang dipicu oleh permintaan pelanggan ataupun kesalahan yang ditemukan saat penggunaan sistem secara *real-time*.

2.8 Unified Modelling Language (UML)

Unified Modelling Language (UML) merupakan sebuah bahasa yang menjadi standar untuk visualisasi, merancang serta mendokumentasikan sebuah sistem perangkat lunak. UML adalah sebuah tools yang digunakan untuk merancang pengembangan software yang berbasis *Object Oriented*. Dengan menggunakan UML kita dapat membuat model untuk perangkat lunak yang akan dibangun.

Terdapat 4 macam hubungan dalam UML, antara lain:

1. *Dependency*

Dependency merupakan hubungan semantic antara dua objek, dimana ketika salah satu benda berubah, maka akan mengakibatkan objek yang lain berubah. *Dependency* digambarkan dengan garis panah putus-putus.

2. *Association*

Association merupakan hubungan antara satu objek dengan objek lainnya yang digambarkan dengan garis serta sebuah label, nama, dan status hubungannya

3. *Generalization*

Generalization merupakan hubungan dimana suatu objek child berbagi tingkah laku dan struktur data dengan objek yang berada di atasnya (parent). Digambarkan dengan garis panah yang menunjuk ke objek parent.

4. *Realization*

Operasi yang benar-benar dilakukan oleh suatu objek UML terdiri atas diagram-diagram sistem dengan sudut pandang tertentu, dimana diagram merupakan sesuatu yang menggambarkan permasalahan ataupun solusi berdasarkan permasalahan suatu model. Diagram yang terdapat pada UML antara lain: *Usecase*, *class diagram*, *state*, *Sequence*, dan *activity*.

2.9 Pengujian

Pengujian merupakan proses dalam kegiatan pengembangan sistem yang dilakukan untuk menemukan kesalahan-kesalahan yang terjadi pada sistem tersebut. Pada pelaksanaannya pengujian dilakukan dengan menggunakan kasus tes yang telah disusun sebelumnya. Kasus tes yang baik adalah yang memiliki kemungkinan menemukan kesalahan pada sistem, sedangkan kasus tes yang sukses adalah yang dapat membongkar kesalahan sistem (Mustaqbal, et al., 2015). Terdapat beberapa jenis pengujian, diantaranya adalah *Black Box Testing*, *White Box Testing*, *Usability Testing*.

2.9.1 *Black Box Testing*

Black Box Testing merupakan pengujian sistem yang fokus pada fungsional dari perangkat lunak. Dalam pelaksanaannya *black box testing* dilakukan dengan mendefinisikan kumpulan kondisi masukan sesuai dengan fungsional dari sistem yang akan digunakan oleh penguji dalam melakukan pengujian pada setiap fungsional sistem (Mustaqbal, et al., 2015).

Black Box Testing dilakukan untuk mengetahui hal-hal berikut :

1. Fungsi yang tidak benar atau tidak ada.
2. Kesalahan antarmuka (*interface errors*).
3. Kesalahan pada struktur data dan akses basis data.
4. Kesalahan performansi (*performance errors*).
5. Kesalahan inisialisasi dan terminasi.

2.9.2 *White Box Testing*

White Box Testing merupakan pengujian sistem yang fokus pada detail perancangan program sistem. Dalam pelaksanaannya *white box testing* menggunakan struktur kontrol dari perancangan program secara prosedural untuk membagi pengujian menjadi beberapa kasus uji. Pengujian ini secara sekilas merupakan pengujian untuk menghasilkan program sistem yang benar secara 100% (Mustaqbal, et al., 2015). Cara pengujian yang dilakukan untuk *white box testing* adalah dengan cara melakukan pengecekan pada setiap fungsi kode program yang ada. Kasus uji yang digunakan dalam *white box testing* antara lain adalah dengan beberapa tahapan yaitu (Mustaqbal, et al., 2015):

1. Pengujian seluruh keputusan yang menggunakan logikal.
2. Pengujian keseluruhan loop yang ada sesuai batasan-batasannya.
3. Pengujian pada struktur data yang sifatnya internal dan yang terjamin validitasnya.

2.9.3 Usability Testing

Usability adalah kemampuan sebuah perangkat lunak untuk menyelesaikan suatu masalah yang dimiliki oleh penggunanya untuk menyelesaikan tugas-tugas tertentu (Aynayya, et al., 2018). Terdapat 5 atribut *Usability* yang digunakan dalam menentukan kualitas dari suatu sistem seperti yang dikemukakan oleh Jacob Nielsen, yaitu:

- a. Mudah untuk dipelajari (*Learnability*).
- b. Efisiensi dalam penggunaan (*Efficiency*).
- c. Mudah untuk diingat (*Memorability*).
- d. Kesalahan dan keamanan (*Error and Safety*).
- e. Kepuasan pengguna (*Satisfaction*).

2.9.3.1 WEBQUAL 4.0

WebQual adalah suatu metode pengukuran kualitas suatu *website* yang diukur dari sisi pengguna. *WebQual* sudah dikembangkan sejak tahun 1998 dan mengalami beberapa perkembangan dari sisi penyusunan dimensi dan pertanyaan-pertanyaan yang diajukan (Sanjaya, 2012). Pada *WebQual* 4.0 tersusun berdasarkan pada penelitian tiga area (dimensi) kualitas seperti yang dijelaskan pada Tabel 2.3 sampai dengan tabel 2.5 (Sanjaya, 2012).

Tabel 2.3 Dimensi Kemudahan Penggunaan (*Usability*)

| No. | Deskripsi Indikator |
|-----|--|
| 1. | Pengguna merasa mudah untuk mempelajari pengoperasian <i>website</i> |
| 2. | Interaksi antara <i>website</i> dengan pengguna jelas dan mudah dipahami |
| 3. | Pengguna merasa mudah untuk bernavigasi dalam <i>website</i> |
| 4. | Pengguna merasa <i>website</i> mudah untuk digunakan |
| 5. | <i>Website</i> memiliki tampilan yang menarik |
| 6. | Desain sesuai dengan jenis <i>website</i> |
| 7. | <i>Website</i> mengandung kompetensi |
| 8. | <i>Website</i> menciptakan pengalaman positif bagi pengguna |

Tabel 2.4 Dimensi Kualitas Interaksi (*Information quality*)

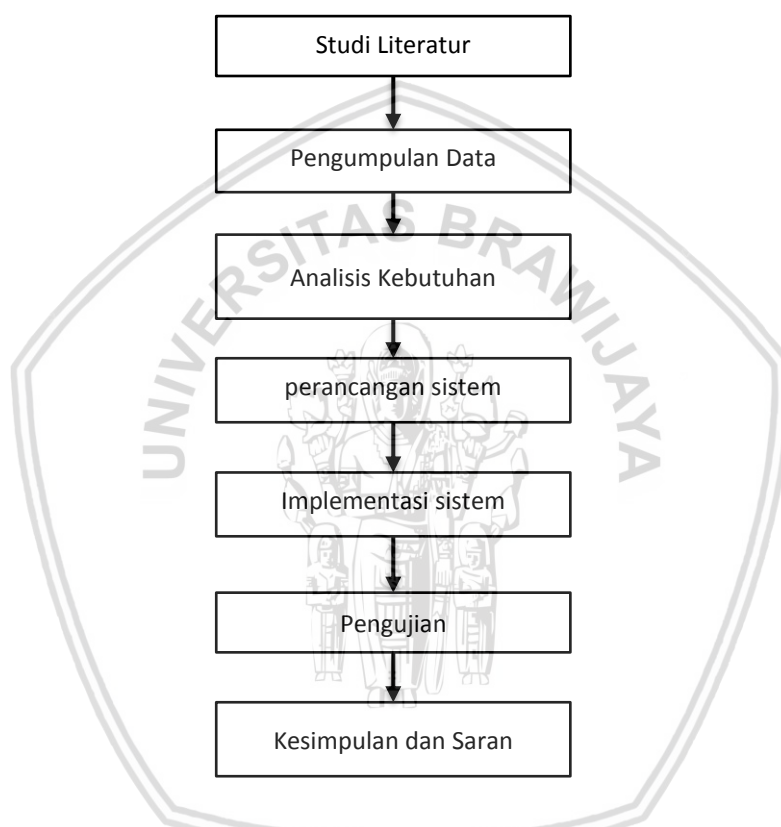
| No. | Deskripsi Indikator |
|-----|--|
| 1. | Pengguna merasa mudah untuk mempelajari pengoperasian <i>website</i> |
| 2. | Interaksi antara <i>website</i> dengan pengguna jelas dan mudah dipahami |
| 3. | Pengguna merasa mudah untuk bernavigasi dalam <i>website</i> |
| 4. | Pengguna merasa <i>website</i> mudah untuk digunakan |
| 5. | <i>Website</i> memiliki tampilan yang menarik |
| 6. | Desain sesuai dengan jenis <i>website</i> |
| 7. | <i>Website</i> mengandung kompetensi |
| 8. | <i>Website</i> menciptakan pengalaman positif bagi pengguna |

Tabel 2.5 Dimensi Kualitas Interaksi (*Interction quality*)

| No. | Deskripsi Indikator |
|-----|--|
| 1. | Website memiliki reputasi yang baik |
| 2. | Pengguna merasa aman untuk melakukan transaksi |
| 3. | Pengguna merasa aman terhadap informasi pribadinya |
| 4. | Website memberi ruang untuk personalisasi |
| 5. | Website memberikan ruang untuk komunitas |
| 6. | Website memberikan kemudahan untuk berkomunikasi dengan organisasi |
| 7. | Pengguna merasa yakin bahwa barang/jasa akan dikirim sebagaimana yang telah dijanjikan |

BAB 3 METODOLOGI PENELITIAN

Metode penelitian dalam penelitian ini dibagi dalam beberapa tahap, yaitu pengumpulan studi literatur, melakukan pengumpulan data dengan cara melakukan wawancara pada masing-masing unit kerja yaitu unit kerja inovasi dan pengembangan manajemen, melakukan analisis kebutuhan, perancangan sistem, pengujian, analisis dan kesimpulan. Berikut tahapan-tahapan dalam penelitian ini ditunjukkan pada Gambar 3.1:



Gambar 3.1 Tahapan Penelitian.

3.1 Studi Literatur

Studi literatur yang dilakukan adalah mengumpulkan dasar-dasar teori yang akan digunakan untuk menunjang penelitian ini. Dasar teori adalah teori-teori yang digunakan sebagai landasan dalam penyusunan sistem ini. Teori-teori tersebut dikumpulkan dari berbagai macam sumber, yaitu melalui buku, jurnal ilmiah, dan media lain seperti portal berita *online*.

3.2 Pengumpulan Data

Pengumpulan data dilakukan dengan cara wawancara. Data yang digunakan adalah data sekunder. Data sekunder adalah data yang didapatkan dari sumber atau literatur. Data tersebut didapatkan dari Departemen Inovasi dan Pengembangan Manajemen PT. Pupuk Kaltim. Data yang dibutuhkan dalam penelitian ini adalah:

1. Data diri dari masing-masing karyawan yang akan mengikuti PIA (Pupuk Kaltim *Innovation Award*) 2017 kurang lebih berjumlah 2103 karyawan.
2. Data struktur organisasi perusahaan yang terdiri dari 6 Direktorat, 21 Kompartement, 88 Departement, dan 845 Bagian.
3. Data prestasi karyawan dalam ajang PIA (Pupuk *Innovation Award*) dan pelatihan yang pernah diikuti.
4. Data peserta dan makalah yang terkumpul pada PIA tahun 2017.
5. Data pelatihan karyawan pada tahun 2015, 2016, 2017.

Selain itu, wawancara dilakukan untuk menggali permasalahan serta kebutuhan sistem serta proses bisnis kegiatan PIA yang sudah dijalankan secara manual. Wawancara dilakukan dengan narasumber yaitu *staff* Departemen Inovasi dan Pengembangan Manajemen PT. Pupuk Kaltim yang bertugas sebagai pengelola kegiatan PIA.

3.3 Analisis Kebutuhan

Analisis kebutuhan sistem dilakukan untuk mengetahui kebutuhan apa saja yang harus ada dalam sistem dengan cara memodelkan proses bisnis saat ini dan usulan proses bisnis. Dari pemodelan tersebut akan mendapatkan prasyarat fungsional dan non-fungsional sistem yang akan dibangun. Menentukan actor yang terkait dengan sistem, menentukan spesifikasi kebutuhan sistem, dan kemudian melakukan pemodelan *use case diagram* dan *use case scenario*.

3.4 Perancangan Sistem

Perancangan sistem adalah tahap yang dilakukan setelah analisis kebutuhan yang bertujuan untuk mendapatkan referensi yang akan digunakan untuk merancang arsitektur dalam tahap implementasi. Dalam tahap ini dilakukan perancangan Arsitektur, *sequence diagram*, perancangan *class diagram*, perancangan data model, perancangan algoritme dan perancangan antarmuka.

3.5 Implementasi Sistem

Implementasi dari sistem dilakukan dengan acuan perancangan aplikasi yang telah disusun sebelumnya. Implementasi sistem ini dibuat dengan berbasis *website* dengan framework *CodeIgniter* dan *MySQL* digunakan sebagai sistem basis data dari *website*. Ada beberapa implementasi dalam pembuatan sistem ini seperti:

1. Implementasi Basis Data
Tujuan dari implementasi basis data ini adalah untuk menyimpan data, dan lain-lain yang masih berhubungan dengan penyimpanan kebutuhan sistem. Basis data yang akan digunakan adalah basis data MySQL.
2. Implementasi *Source Code*
Bagian ini berguna untuk menerjemahkan dari perancangan kedalam bahasa pemrograman. Seluruh proses implementasi sistem dan perhitungan dilakukan oleh bagian ini. Implementasi *source code* dilakukan menggunakan notepad++ dengan bahasa PHP dan *framework CodeIgniter*.
3. Implementasi Antarmuka
Implementasi antarmuka menunjukkan hasil implementasi antarmuka sistem yang telah dibangun berdasarkan pada perancangan antarmuka yang telah dirancang pada tahap perancangan antarmuka sistem.

3.6 Pengujian

Pengujian dilakukan dengan cara melakukan pengujian dengan metode *whitebox* yang mana pada metode ini akan menguji alur dari kode program yang telah diimplementasikan sehingga akan diketahui keberhasilan dari kode program tersebut. Pada pengujian *whitebox* dilakukan dengan melakukan pengujian unit dan pengujian integrasi. Terdapat pula pengujian lain yaitu pengujian validasi dengan metode *blackbox* untuk melihat ke arah fungsionalitas sistem. Pada pengujian ini akan dilihat kesesuaian kinerja sistem dengan kebutuhan pengguna. Jika hasil keluaran dan yang diinginkan sesuai maka fungsi tersebut valid. Pengujian validasi dengan *blackbox* ini hanya melihat keluaran dari sistem tanpa memperhatikan alur dari kode program. Selain itu akan dilakukan juga pengujian *usability* untuk mengetahui tingkat keberhasilan sistem ditinjau dari pengujian yang dilakukan langsung oleh pengguna.

3.7 Kesimpulan dan Saran

Bagian kesimpulan dilakukan setelah melalui bagian pendahuluan, tahap analisis dan perancangan, implementasi serta pengujian dan didasarkan pada kesesuaian teori. Kesimpulan diambil untuk menjawab rumusan masalah yang telah didefinisikan sebelumnya. Kemudian saran diberikan guna memperbaiki kesalahan dan untuk keperluan pengembangan lebih lanjut.

BAB 4 ANALISIS KEBUTUHAN

Bab ini menjelaskan tentang analisis kebutuhan dari sistem informasi penjurian Online pada Pupuk Kaltim *Inovation Award*. Analisis kebutuhan dimulai dengan menjelaskan penjabarkan gambaran umum sistem, setelah itu dilanjutkan dengan mengidentifikasi actor, dan merumuskan kebutuhan fungsional dan non fungsional sistem.

1.1 Gambaran Umum Sistem

Sistem yang akan dikembangkan dalam penelitian ini adalah sistem penjurian *Online* berbasis web yang dapat membantu proses kegiatan PIA menjadi lebih efisien dan terintegrasi. Pada sistem ini tim telaah dapat mengkaji setiap makalah yang terkumpul dengan lebih cepat dan akurat dengan adanya fitur cek plagiasi. Pada fitur cek plagiasi tim telaah dapat memberikan masukan berupa text abstrak dari makalah yang ingin dicek kemudian sistem akan mencocokkan data tersebut dengan data abstrak dari makalah terdahulu yang tersimpan didalam basis data sistem kemudian sistem akan memberikan nilai tingkat plagiasi. Sistem juga dapat membantu pengelola kegiatan dapat mengelola kegiatan dengan data yang sudah terintegrasi pada sistem serta memiliki fitur rekomendasi juri tetap sehingga pengelola dapat menetapkan juri tetap untuk kegiatan PIA. Selain itu, sistem juga menyediakan kemudahan bagi seorang juri untuk melakukan penilaian yaitu dengan cara menjadikan *form* penilaian pada sitem secara interaktif. *Form* didesain secara interaktif dengan menampilkan segala informasi pada setiap *range* angka pada saat pointer ditunjukkan pada *range* angka tersebut. Selain itu pada fitur penilaian ini juga menyediakan perhitungan hasil akhir secara otomatis sehingga juri tidak perlu menghitung secara manual.

4.1 Identifikasi Actor

Indentifikasi *actor* adalah menentukan seseorang atau sistem lain yang terlibat dengan sistem secara langsung. Tahap ini dilakukan berdasarkan analisis kebutuhan yang telah dilakukan sebelumnya yang dideskripsikan pada Tabel 4.1.

Tabel 4.1 Identifikasi Actor

| Actor | Deskripsi |
|---------------|--|
| Administrator | Karyawan yang bertugas sebagai pengelola kegiatan dan menggunakan sistem untuk mengelola data pengguna sistem lain. |
| Juri | Karyawan yang bertugas sebagai juri dan menggunakan sistem untuk melakukan penilaian. |
| Tim telaah | Karyawan yang bertugas untuk mengkaji makalah dan menggunakan sistem untuk melakukan cek plagiasi setiap makalah yang terkumpul. |

4.2 Daftar Kebutuhan Sistem

Kebutuhan fungsional merupakan suatu layanan yang harus disediakan oleh sistem. Spesifikasi kebutuhan juri ditunjukkan pada Tabel 4.2, spesifikasi kebutuhan tim telaah ditunjukkan pada Tabel 4.3, dan spesifikasi kebutuhan administrator ditunjukkan pada Tabel 4.4. Setiap kebutuhan akan diberikan kode SIPO_XXYY. SIPO merupakan singkatan dari Sistem Informasi Penjurian Online sedangkan XX menunjukkan nomor kebutuhan fungsional atau non fungsional. Untuk kode kebutuhan fungsional adalah 01, sedangkan untuk kebutuhan non fungsional adalah 2. Kode YY menunjukkan nomor dari kebutuhan utama.

4.2.1 Daftar Kebutuhan Fungsional

Tabel 4.2 Spesifikasi Kebutuhan Juri

| No. | Kode | Nama fungsi | Deskripsi |
|-----|-----------|------------------------------------|---|
| 1. | SIPO_0101 | Login | Sistem harus menyediakan fungsi <i>login</i> untuk mengizinkan pengguna mengakses sistem sesuai dengan otoritasnya dengan memasukkan nama pengguna dan kata sandi. |
| 2. | SIPO_0102 | Menilai makalah | <p>Sistem harus menyediakan fungsi menilai makalah untuk seorang juri agar dapat melakukan penilaian.</p> <p>Spesifikasi kebutuhan :</p> <ul style="list-style-type: none"> • Sistem harus menyediakan <i>form</i> penilaian sesuai dengan format <i>form</i> yang menampilkan setiap kriteria penilaian yang mana pada setiap kriteria juri dapat mengisikan nilainya dengan cara memilih angka yang telah disajikan. (SIPO_01021) • <i>Form</i> dapat menampilkan <i>pop-up</i> kriteria angka ketika kursor diletakkan pada angka yang dipilih. (SIPO_01022) • Sistem harus dapat mengakumulasikan nilai setiap kriteria menjadi sebuah nilai akhir dengan rumus perhitungan yang sudah ditetapkan. (SIPO_01023) |
| 3. | SIPO_0103 | Melihat data makalah <i>stream</i> | Sistem harus menyediakan fungsi melihat data makalah <i>stream</i> dalam bentuk tabel. |

| | | | |
|----|-----------|------------------------------|---|
| | | | Spesifikasi kebutuhan : <ul style="list-style-type: none"> Data yang ditampilkan adalah data makalah sesuai <i>stream</i> yang telah dibagi dan tersedia tombol <i>download</i> dan nilai untuk setiap makalah. (SIPO_01031) |
| 4. | SIPO_0104 | Melihat hasil penilaian juri | Sistem harus menyediakan fungsi melihat hasil penilaian juri dalam bentuk tabel. Spesifikasi Kebutuhan : <ul style="list-style-type: none"> Data yang ditampilkan adalah judul makalah dan nilai setiap juri. (SIPO_01041) |
| 5. | SIPO_0105 | Melihat nilai akhir | Sistem harus menyediakan fungsi melihat nilai akhir makalah setiap <i>stream</i> dalam bentuk tabel. Spesifikasi kebutuhan : <ul style="list-style-type: none"> Data yang ditampilkan adalah data makalah setiap <i>stream</i> dan nilai akhir yang sudah diakumulasikan dari nilai semua juri yang memiliki <i>stream</i> yang sama. (SIPO_01051) |
| 6. | SIPO_0106 | Mengunduh makalah | Sistem harus dapat menyediakan fungsi mengunduh data makalah yang ditampilkan. |
| 7. | SIPO_0107 | Logout | Sistem harus menyediakan fungsi <i>logout</i> untuk keluar dari sistem. |

Tabel 4.3 Spesifikasi Kebutuhan Tim Telaah

| No. | Kode | Nama fungsi | Deskripsi |
|-----|-----------|--------------|--|
| 1. | SIPO_0101 | Login | Sistem harus menyediakan fungsi <i>login</i> untuk mengizinkan pengguna mengakses sistem sesuai dengan otoritasnya dengan memasukkan nama pengguna dan kata sandi. |
| 2. | SIPO_0108 | Cek plagiasi | Sistem harus menyediakan fungsi cek plagiasi untuk mendapatkan tingkat plagiasi sebuah makalah. |

| | | | |
|----|-----------|------------------------------|---|
| | | | Spesifikasi kebutuhan: <ul style="list-style-type: none"> Data yang harus dimasukkan adalah bagian abstrak dari makalah. (SIPO_01081) Sistem harus dapat menampilkan hasil tingkat plagiasi dalam bentuk persen. (SIPO_01082) |
| 3. | SIPO_0109 | Melihat makalah | Sistem harus menyediakan fungsi melihat daftar makalah yang akan ditelaah. Spesifikasi kebutuhan : <ul style="list-style-type: none"> Data yang ditampilkan adalah data makalah yang terkumpul dan tombol <i>download</i> untuk setiap makalah. (SIPO_01091) |
| 4. | SIPO_0110 | Menentukan kelolosan makalah | Sistem harus menyediakan fungsi menentukan kelolosan gugus untuk menentukan gugus yang lolos dalam tahap telaah makalah oleh tim telaah. |
| 5. | SIPO_0106 | Mengunduh makalah | Sistem harus dapat menyediakan fungsi mengunduh data makalah yang ditampilkan. |
| 6. | SIPO_0107 | Logout | Sistem harus menyediakan fungsi <i>logout</i> untuk keluar dari sistem. |

Tabel 4.4 Spesifikasi Kebutuhan Administrator

| No. | Kode | Nama fungsi | Deskripsi |
|-----|-----------|------------------|--|
| 1. | SIPO_0101 | Login | Sistem harus menyediakan fungsi <i>login</i> untuk mengizinkan pengguna mengakses sistem sesuai dengan otoritasnya dengan memasukkan nama pengguna dan kata sandi. |
| 2. | SIPO_0111 | Rekomendasi juri | Sistem harus menyediakan fungsi rekomendasi juri sesuai dengan persyaratan yang ada. Spesifikasi kebutuhan : <ul style="list-style-type: none"> Juri adalah karyawan yang sudah mengikuti pelatihan juri pemula. (SIPO_01111) |

| | | | |
|----|-----------|----------------------------------|--|
| | | | <ul style="list-style-type: none"> Prioritas juri yang direkomendasikan adalah karyawan dengan pelatihan terbanyak. (SIPO_01112) |
| 3. | SIPO_0112 | Melihat data calon juri | <p>Sistem harus menyediakan fungsi melihat data calon juri dalam bentuk tabel.</p> <p>Spesifikasi kebutuhan :</p> <ul style="list-style-type: none"> Data yang ditampilkan pada tabel berupa data karyawan dan pelatihan yang pernah diikuti. (SIPO_01121) |
| 4. | SIPO_0113 | Memilih juri tetap | Sistem harus menyediakan fungsi memilih juri tetap setelah sistem memberikan rekomendasi juri. |
| 5. | SIPO_0114 | Menentukan <i>stream</i> juri | Sistem harus menyediakan fungsi menentukan <i>stream</i> juri sehingga juri hanya akan menilai makalah sesuai <i>stream</i> yang ditentukan. |
| 6. | SIPO_0115 | Melihat juri tetap | <p>Sistem harus menyediakan fungsi melihat juri tetap dalam bentuk tabel.</p> <p>Spesifikasi kebutuhan :</p> <ul style="list-style-type: none"> Data yang ditampilkan adalah data juri tetap yang telah dipilih oleh administrator dan terdapat kolom <i>stream</i> untuk menampilkan <i>stream</i> juri. (SIPO_01151) |
| 7. | SIPO_0116 | Menghapus juri tetap | Sistem harus menyediakan fungsi menghapus juri tetap. |
| 8. | SIPO_0117 | Menentukan <i>stream</i> makalah | Sistem harus menyediakan fungsi menentukan <i>stream</i> makalah sehingga makalah hanya akan dinilai oleh juri dengan <i>stream</i> yang sama. |
| 9. | SIPO_0118 | Melihat data makalah terdaftar | <p>Sistem harus menyediakan fungsi melihat makalah yang terdaftar.</p> <p>Spesifikasi kebutuhan :</p> <ul style="list-style-type: none"> Data yang ditampilkan adalah data makalah yang terdaftar |

| | | | |
|-----|-----------|-----------------------------------|---|
| | | | serta kolom tim telaah untuk menentukan tim telaah yang akan mengkaji makalah. (SIPO_01181) |
| 10. | SIPO_0119 | Menentukan tim telaah makalah | Sistem harus menyediakan fungsi menentukan tim telaah sehingga makalah hanya dikaji oleh seorang tim telaah saja. |
| 11. | SIPO_0120 | Melihat data makalah lolos | Sistem harus menyediakan fungsi melihat makalah yang telah lolos dalam bentuk tabel. Spesifikasi kebutuhan : <ul style="list-style-type: none"> Data yang ditampilkan adalah data makalah yang telah lolos tahap penkajian dari tim telaah, kolom <i>stream</i> untuk menampilkan <i>stream</i> makalah, dan tombol <i>download</i> untuk setiap makalah. (SIPO_01201) |
| 12. | SIPO_0121 | Melihat data tim telaah | Sistem harus menyediakan fungsi melihat data tim telaah dalam bentuk tabel. |
| 13. | SIPO_0122 | Menambah data tim telaah | Sistem harus menyediakan fungsi menambah tim telaah. |
| 14. | SIPO_0123 | Menghapus data tim telaah | Sistem harus menyediakan fungsi menghapus tim telaah. |
| 15. | SIPO_0124 | Melihat data pelatihan calon juri | Sistem harus menyediakan fungsi melihat data calon juri dalam bentuk tabel. |
| 16. | SIPO_0125 | Menambah pelatihan calon juri | Sistem harus menyediakan fungsi menambah pelatihan calon juri. |
| 15. | SIPO_0106 | Mengunduh makalah | Sistem harus dapat menyediakan fungsi mengunduh data makalah yang ditampilkan. |
| 17. | SIPO_0107 | Logout | Sistem harus menyediakan fungsi <i>logout</i> untuk keluar dari sistem. |

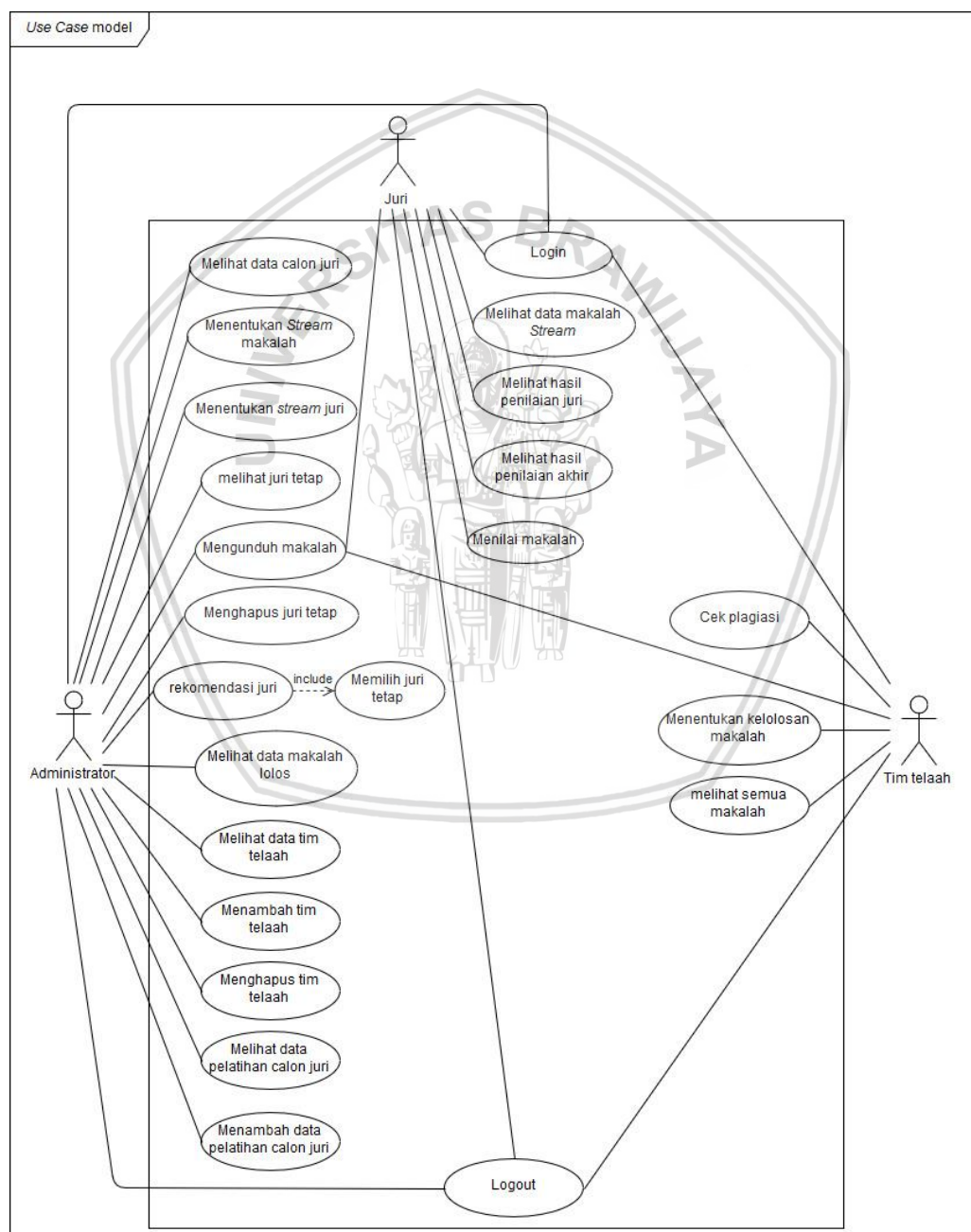
4.2.2 Daftar Kebutuhan Non Fungsional

Tabel 4.5 Kebutuhan Non Fungsional Sistem

| No. | Kode | Nama Kebutuhan | Deskripsi |
|-----|-----------|------------------|--|
| 1. | SIPO_0201 | <i>Usability</i> | Sistem harus memiliki tampilan yang mudah digunakan oleh pengguna. |

4.3 Pemodelan Kebutuhan

4.3.1 Pemodelan Use Case Diagram



Gambar 4.1 Use Case Diagram SIPO.

4.4 Use Case Scenario

Use case scenario menerangkan alur dari setiap *use case diagram* sistem yang telah didefinisikan pada Gambar 4.1. *Use case scenario* digambarkan dalam bentuk tabel yang terdiri dari *actor*, *objective*, *pre-condition*, *main flow*, *alternative flow*, *post-condition*.

4.4.1 Login

Tabel 4.6 Use Case Scenario Login

| | |
|-------------------------|--|
| Actor | Administrator, Juri, Tim Telaah. |
| Objective | Mengizinkan actor mengakses sistem sesuai dengan otoritas masing-masing. |
| Pre-Condition | Actor telah membuka <i>form</i> Login. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memasukkan nama pengguna dan kata sandi lalu memilih tombol Login. 2. Sistem menampilkan halaman awal sesuai dengan status masing-masing. |
| Alternative Flow | <ol style="list-style-type: none"> 1. Jika Actor memilih tombol Login tetapi nama pengguna dan kata sandi masih kosong maka akan menampilkan pesan "Harap isi bidang" pada kolom yang kosong. 2. Jika nama pengguna dan kata sandi salah maka akan menampilkan pesan "Password/username yang anda masukkan salah". |
| Post-Condition | Actor dapat mengakses sistem sesuai dengan otoritas masing-masing. |

4.4.2 Menilai Makalah

Tabel 4.7 Use Case Scenario Menilai Makalah

| | |
|-------------------------|--|
| Actor | Juri |
| Objective | Mengizinkan actor untuk melihat hasil akhir penilaian. |
| Pre-Condition | Actor telah membuka <i>form</i> penilaian. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memberi nilai pada setiap kategori penilaian yang ada dengan cara memilih nominal angka yang berada pada <i>form</i> penilaian. 2. Actor memilih tombol nilai. |
| Alternative Flow | <ol style="list-style-type: none"> 1. Jika Actor memilih tombol Nilai tetapi masih ada kolom penilaian yang belum diisi maka akan menampilkan pesan "Harap isi bidang" pada kolom yang kosong. |

| | |
|-----------------------|---|
| Post-Condition | Actor berhasil melihat hasil akhir penilaian. |
|-----------------------|---|

4.4.3 Melihat Data Makalah *Stream*

Tabel 4.8 Use Case Scenario Melihat Data Makalah *Stream*

| | |
|-------------------------|---|
| Actor | Juri. |
| Objective | Mengizinkan actor untuk melihat data makalah sesuai <i>stream</i> . |
| Pre-Condition | Actor telah masuk ke dalam sistem sebagai juri. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih menu makalah. 2. Sistem menampilkan data makalah sesuai <i>stream</i> pada halaman makalah. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil melihat data makalah sesuai <i>stream</i> . |

4.4.4 Melihat Hasil Penilaian Juri

Tabel 4.9 Use Case Scenario Melihat Hasil Penilaian Juri

| | |
|-------------------------|---|
| Actor | Juri. |
| Objective | Mengizinkan actor melihat hasil penilaian yang dilakukan actor. |
| Pre-Condition | Actor telah masuk ke dalam sistem sebagai juri. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih menu makalah. 2. Sistem menampilkan data makalah sesuai <i>stream</i> pada halaman makalah beserta nilai dari setiap makalah. |
| Alternative Flow | <ol style="list-style-type: none"> 1. Jika actor belum melakukan penilaian pada suatu makalah makal kolom nilai berisi tombol untuk melakukan penilaian. |
| Post-Condition | Actor berhasil melihat hasil penilaian makalah yang telah dinilai oleh actor. |

4.4.5 Melihat Nilai Akhir

Tabel 4.10 Use Case Scenario Melihat Nilai Akhir

| | |
|----------------------|---|
| Actor | juri. |
| Objective | Mengizinkan actor melihat hasil akhir penilaian akumulasi nilai yang dilakukan semua juri dengan <i>stream</i> yang sama. |
| Pre-Condition | Actor telah masuk ke dalam sistem sebagai juri. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih menu Hasil Akhir. |

| | |
|-------------------------|---|
| | 2. Sistem menampilkan data makalah sesuai stream pada halaman nilai akhir beserta nilai akhirnya. |
| Alternative Flow | 1. Jika makalah belum dinilai maka kolom nilai akhir akan kosong. |
| Post-Condition | Actor berhasil melihat nilai akhir dari setiap makalah. |

4.4.6 Mengunduh Makalah

Tabel 4.11 Use Case Scenario Mengunduh Makalah

| | |
|-------------------------|---|
| Actor | Tim telaah, juri, admin. |
| Objective | Mengizinkan actor menyimpan makalah pada komputer. |
| Pre-Condition | Actor telah membuka halaman makalah. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih tombol download pada makalah yang ingin diunduh. 2. Sistem menyimpan makalah pada komputer. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil menyimpan makalah pada komputer. |

4.4.7 Logout

Tabel 4.12 Use Case Scenario Logout

| | |
|-------------------------|---|
| Actor | Administrator, Juri, Tim Telaah. |
| Objective | Mengizinkan actor untuk keluar dari sistem. |
| Pre-Condition | Actor masih berada dalam sistem sesuai dengan otoritas masing-masing. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih menu logout. 2. Sistem menghapus <i>session Login</i>, kemudian sistem menampilkan halaman utama. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil keluar sistem. |

4.4.8 Cek Plagiasi

Tabel 4.13 Use Case Scenario Cek Plagiasi

| | |
|----------------------|---|
| Actor | Tim Telaah |
| Objective | Mengizinkan Actor untuk melihat tingkat plagiasi dari abtrak yang dimasukkan. |
| Pre-Condition | Actor telah membuka halaman cek plagiasi. |

| | |
|-------------------------|--|
| Main Flow | <ol style="list-style-type: none"> 1. Actor memasukkan abstrak yang akan diuji kedalam kolom abstrak pada halaman cek plagiasi lalu actor memilih tombol cek. 2. Sistem menampilkan tingkat plagiasi dari abstrak yang dimasukkan. |
| Alternative Flow | <ol style="list-style-type: none"> 1. Jika Actor memilih tombol cek tetapi kolom abstrak masih belum diisi maka akan menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Post-Condition | Actor berhasil melihat tingkat plagiasi dari abstrak yang dimasukkan. |

4.4.9 Melihat Makalah

Tabel 4.14 Use Case Scenario Melihat Makalah

| | |
|-------------------------|---|
| Actor | Tim Telaah. |
| Objective | Mengizinkan actor melihat semua makalah yang sudah terdaftar. |
| Pre-Condition | Actor telah masuk ke dalam sistem sebagai tim telaah. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih menu makalah. 2. Sistem menampilkan semua makalah yang terdaftar. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil melihat semua makalah yang telah terdaftar. |

4.4.10 Menentukan Kelolosan Makalah

Tabel 4.15 Use Case Scenario Menentukan Kelolosan Makalah

| | |
|-------------------------|---|
| Actor | Tim Telaah. |
| Objective | Mengizinkan actor menentukan kelolosan gugus. |
| Pre-Condition | Actor telah berada pada halaman edit informasi makalah. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih keterangan lolos/tidak lolos pada kolom status. 2. Actor memilih tombol simpan. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil menentukan kelolosan gugus. |

4.4.11 Rekomendasi Juri

Tabel 4.16 Use Case Scenario Rekomendasi Juri

| | |
|-------------------------|--|
| Actor | Admin. |
| Objective | Mengizinkan actor untuk melihat rekomendasi juri. |
| Pre-Condition | Actor telah berada pada halaman data calon juri. |
| Main Flow | 1. Actor memilih tombol rekomendasi juri. 2. Sistem menampilkan rekomendasi juri. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil melihat rekomendasi juri. |

4.4.12 Melihat Data Calon Juri

Tabel 4.17 Use Case Scenario melihat Data Calon Juri

| | |
|-------------------------|---|
| Actor | Admin. |
| Objective | Mengizinkan actor untuk melihat data calon juri. |
| Pre-Condition | Actor telah masuk ke dalam sistem dan berada pada halaman makalah. |
| Main Flow | 1. Actor memilih menu rekomendasi juri. 2. Sistem menampilkan halaman data calon juri. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil melihat data calon juri. |

4.4.13 Memilih Juri Tetap

Tabel 4.18 Use Case Scenario Memilih Juri

| | |
|-------------------------|--|
| Actor | Admin. |
| Objective | Mengizinkan actor memilih juri tetap. |
| Pre-Condition | Actor telah membuka data rekomendasi juri. |
| Main Flow | 1. Actor memilih tombol tambah pada juri yang dipilih. |
| Alternative Flow | 1. Apabila pada data rekomendasi juri terdapat juri yang telah terpilih maka pada kolom tambah juri akan menampilkan "juri tetap" dan tidak dapat dipilih kembali. |
| Post-Condition | Actor berhasil memilih juri tetap. |

4.4.14 Menentukan *Stream* Juri

Tabel 4.19 *Use Case Scenario* Menentukan *Stream* Juri

| | |
|-------------------------|--|
| Actor | Admin. |
| Objective | Mengizinkan actor untuk menentukan <i>stream</i> juri. |
| Pre-Condition | Actor telah berada pada halaman. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih tombol edit. 2. Sistem menampilkan formulir dengan kolom <i>stream</i>. 3. Actor mengisi kolom <i>stream</i> dan menekan tombol simpan. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil menentukan <i>stream</i> juri. |

4.4.15 Melihat Juri Tetap

Tabel 4.20 *Use Case Scenario* Melihat Juri Tetap

| | |
|-------------------------|---|
| Actor | Admin. |
| Objective | Mengizinkan actor melihat data juri tetap. |
| Pre-Condition | Actor telah masuk ke dalam sistem dan berada pada halaman makalah. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih menu Juri. 2. Sistem menampilkan data juri tetap pada halaman juri. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil melihat data juri tetap pada halaman juri. |

4.4.16 Menghapus Juri

Tabel 4.21 *Use Case Scenario* menghapus Juri

| | |
|-------------------------|--|
| Actor | Admin. |
| Objective | Mengizinkan actor untuk menghapus juri. |
| Pre-Condition | Actor telah berada pada halaman juri tetap. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih tombol hapus pada juri yang terpilih. 2. Sistem menampilkan <i>pop up</i> dengan pesan “apakah anda yakin?”. 3. Actor memilih tombol yakin. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil menghapus juri yang terpilih. |

4.4.17 Menentukan *Stream* Makalah

Tabel 4.22 *Use Case Scenario* Menentukan *Stream* Makalah

| | |
|-------------------------|---|
| Actor | Admin. |
| Objective | Mengizinkan actor untuk menentukan <i>stream</i> makalah. |
| Pre-Condition | Actor telah berada pada halaman makalah. |
| Main Flow | 4. Actor memilih tombol edit. 5. Sistem menampilkan formulir dengan kolom <i>stream</i> . 6. Actor mengisi kolom <i>stream</i> dan menekan tombol simpan. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil menentukan <i>stream</i> makalah. |

4.4.18 Melihat Data Makalah Terdaftar

Tabel 4.23 *Use Case Scenario* Melihat Makalah Terdaftar

| | |
|-------------------------|---|
| Actor | Admin. |
| Objective | Megizinkan actor melihat semua makalah yang terdaftar. |
| Pre-Condition | Actor telah berada pada halaman makalah. |
| Main Flow | 7. Actor memilih menu makalah terdaftar. 8. Sistem menampilkan semua makalah yang terdaftar. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil melihat semua makalah yang terdaftar. |

4.4.19 Menentukan Tim Telaah Makalah

Tabel 4.24 *Use Case Scenario* Menentukan Tim Telaah Makalah

| | |
|-------------------------|--|
| Actor | Admin. |
| Objective | Mengizinkan actor untuk menentukan tim telaah makalah. |
| Pre-Condition | Actor telah berada pada halaman makalah terdaftar. |
| Main Flow | 1. Actor memilih tombol edit. 2. Sistem menampilkan formulir dengan kolom tim telaah. 3. Actor mengisi kolom tim telaah dan menekan tombol simpan. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil menentukan tim telaah makalah. |

4.4.20 Melihat Data Makalah Lolos

Tabel 4.25 Use Case Scenario Melihat Makalah Lolos

| | |
|-------------------------|---|
| Actor | Admin. |
| Objective | Mengizinkan actor melihat semua makalah yang telah lolos. |
| Pre-Condition | Actor telah masuk ke dalam sistem dan berada pada halaman makalah. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih menu makalah. 2. Sistem menampilkan semua makalah yang telah lolos. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil melihat semua makalah yang telah lolos. |

4.4.21 Melihat Data Tim Telaah

Tabel 4.26 Use Case Scenario Melihat Data Tim Telaah

| | |
|-------------------------|---|
| Actor | Admin. |
| Objective | Mengizinkan actor melihat data tim telaah. |
| Pre-Condition | Actor telah masuk ke dalam sistem dan berada pada halaman makalah. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih menu tim telaah. 2. Sistem menampilkan data tim telaah pada halaman tim telaah. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil melihat data tim telaah. |

4.4.22 Menambah Data Tim Telaah

Tabel 4.27 Use Case Scenario Menambah Data Tim Telaah

| | |
|----------------------|---|
| Actor | Admin. |
| Objective | Mengizinkan actor untuk menambah tim telaah. |
| Pre-Condition | Actor telah berada pada halaman data tim telaah. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih tombol tambah. 2. Sistem menampilkan formulir dengan kolom npk, nama, kata sandi. 3. Actor mengisi kolom npk, nama, kata sandi. 4. Actor memilih tombol tambah. |

| | |
|-------------------------|--|
| Alternative Flow | <ol style="list-style-type: none"> 1. Jika Actor memilih tombol tambah tetapi kolom npk, nama, atau kata sandi masih belum diisi maka akan menampilkan pesan “Harap isi bidang” pada kolom yang kosong. 2. Jika data yang diinputkan sudah terdapat pada data tim telaah maka akan menampilkan pesan “tim telaah telah terpilih” |
| Post-Condition | Actor berhasil menambahkan tim telaah. |

4.4.23 Menghapus Data Tim Telaah

Tabel 4.28 Use Case Scenario Menambah Data Tim Telaah

| | |
|-------------------------|--|
| Actor | Admin. |
| Objective | Mengizinkan actor untuk menghapus tim telaah. |
| Pre-Condition | Actor telah berada pada halaman data tim telaah. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih tombol hapus pada tim telaah yang terpilih. 2. Sistem menampilkan <i>pop up</i> dengan pesan “apakah anda yakin?”. 3. Actor memilih tombol yakin. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil menghapus tim telaah yang terpilih. |

4.4.24 Melihat Data Pelatihan Calon Juri

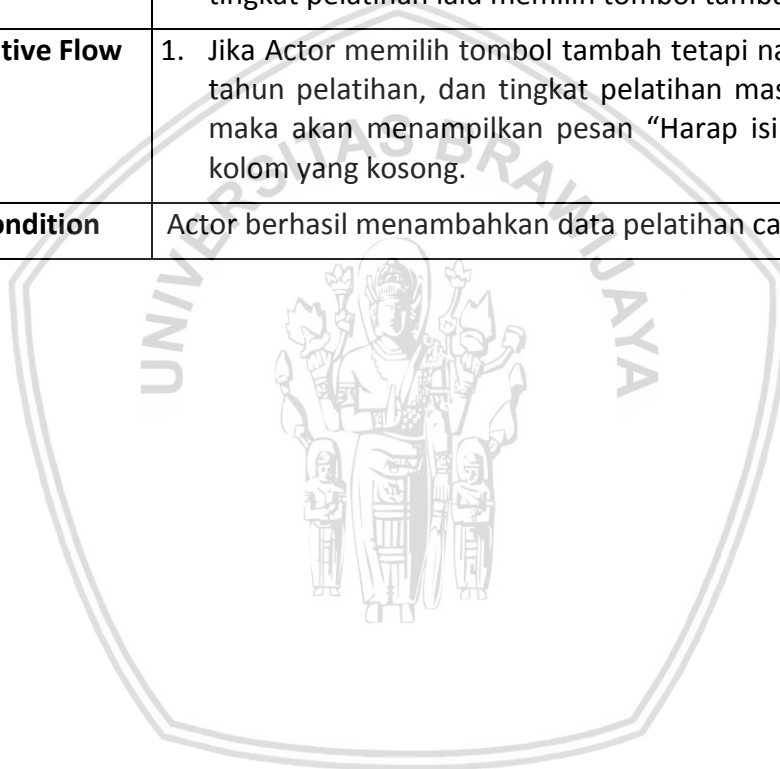
Tabel 4.29 Use Case Scenario Melihat Data Pelatihan Calon Juri

| | |
|-------------------------|--|
| Actor | Admin. |
| Objective | Mengizinkan actor melihat data pelatihan calon juri |
| Pre-Condition | Actor telah berada pada halaman data calon juri. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih tombol lihat pelatihan. 2. Sistem menampilkan data pelatihan calon juri. |
| Alternative Flow | - |
| Post-Condition | Actor berhasil melihat data pelatihan calon juri. |

4.4.25 Menambah Data Pelatihan Calon Juri

Tabel 4.30 *Use Case Scenario* Menambah Data Pelatihan Calon Juri

| | |
|-------------------------|---|
| Actor | Admin. |
| Objective | Mengizinkan actor untuk menambah data pelatihan calon juri. |
| Pre-Condition | Actor telah berada pada halaman pelatihan calon juri. |
| Main Flow | <ol style="list-style-type: none"> 1. Actor memilih tombol tambah. 2. Sistem menampilkan formulir dengan kolom nama pelatihan, tahun pelatihan, dan tingkat pelatihan. 3. Actor mengisi kolom nama pelatihan, tahun pelatihan, dan tingkat pelatihan lalu memilih tombol tambah. |
| Alternative Flow | <ol style="list-style-type: none"> 1. Jika Actor memilih tombol tambah tetapi nama pelatihan, tahun pelatihan, dan tingkat pelatihan masih belum diisi maka akan menampilkan pesan "Harap isi bidang" pada kolom yang kosong. |
| Post-Condition | Actor berhasil menambahkan data pelatihan calon juri. |



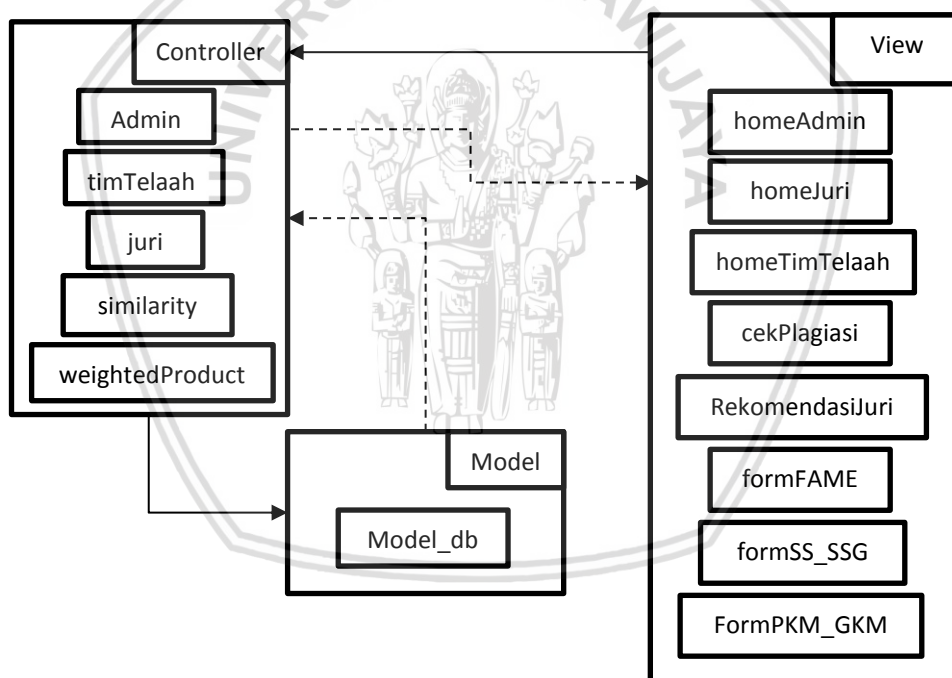
BAB 5 PERANCANGAN DAN IMPLEMENTASI

5.1 Perancangan

Pada tahap perancangan ini akan ditampilkan beberapa diagram perancangan meliputi *sequence diagram*, *class diagram*, perancangan algoritme, serta akan ditampilkan perancangan antarmuka yang nantinya akan digunakan untuk pedoman dalam melakukan implementasi.

5.1.1 Perancangan Arsitektur Sistem

Sistem Penjurian Online dibangun dengan menggunakan arsitektur MVC (*Model View Controller*) dimana *Model* merupakan kelas yang merepresentasikan basis data sistem yang berisi fungsi-fungsi untuk mengakses basis data sistem. *View* merupakan kelas yang akan berinteraksi langsung dengan pengguna. Dan *Controller* merupakan kelas perantara antara *Model* dan *View*. Sistem penjurian online dikembangkan dengan menggunakan *framework Code Igniter* dimana *framework* ini telah bekerja dengan arsitektur MVC.



Gambar 5.1 Diagram Arsitektur MVC

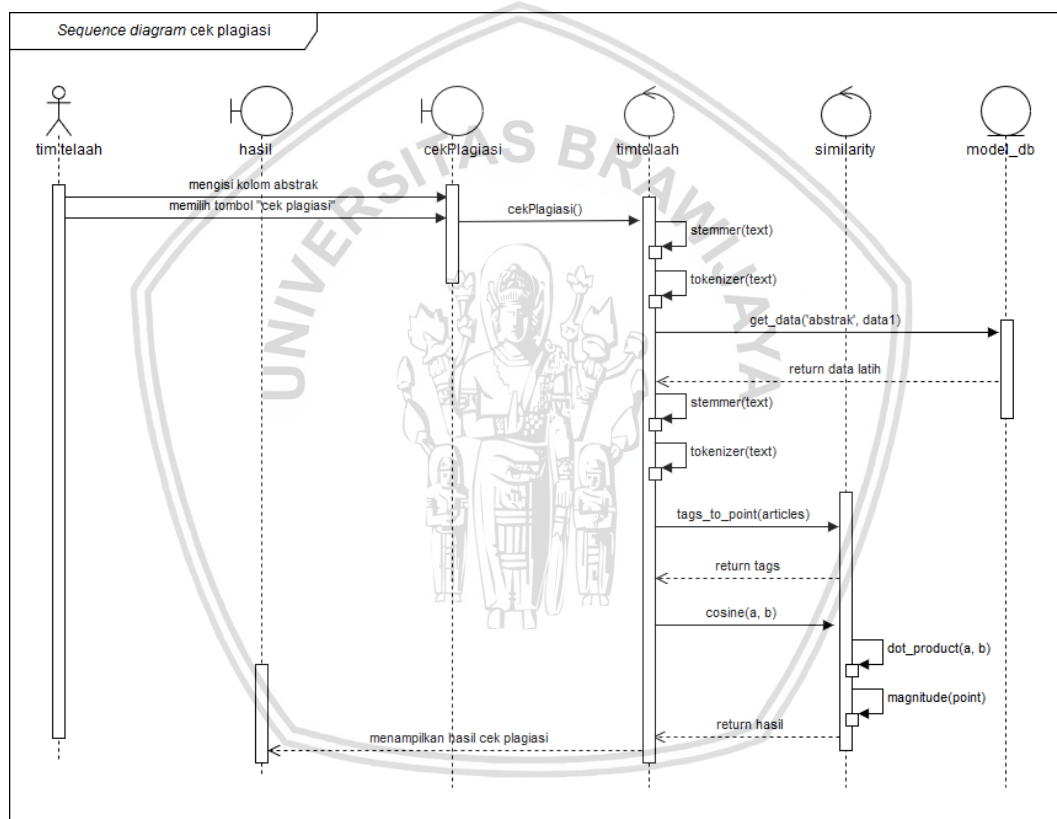
Pada Gambar 5.1 ditunjukkan diagram arsitektur MVC dari sistem setelah dilakukan proses analisis kebutuhan. Pada tahap perancangan ini terdapat kelas admin, timTelaah, juri, similarity, serta weightedProduct yang berperan sebagai kelas controller yang bekerja untuk menghubungkan kelas view dan model, kelas model_db yang berperan sebagai kelas model pada sistem yang bekerja untuk mengakses data-data pada tabel yang ada pada basis data sistem.

Pada sistem penjurian online terdapat kebutuhan-kebutuhan fungsional yang telah didefinisikan pada proses analisis kebutuhan salah satunya adalah fungsional cek plagiasi. Pada fungsional cek plagiasi pada sistem ini menggunakan *library* sastrawi dalam melakukan proses *preprocessing text* yaitu proses *stemming* dan proses tokenisasi.

5.1.2 Sequence Diagram

Pemodelan *sequence diagram* menjelaskan tentang urutan proses yang terjadi pada suatu fungsional sistem. *Sequence diagram* dimodelkan dengan mengacu pada *use case diagram* dan *use case scenario* yang telah dijelaskan sebelumnya. Setiap *sequence diagram* menjelaskan urutan proses suatu *use case*.

5.1.2.1 Sequence Diagram Cek Plagiasi

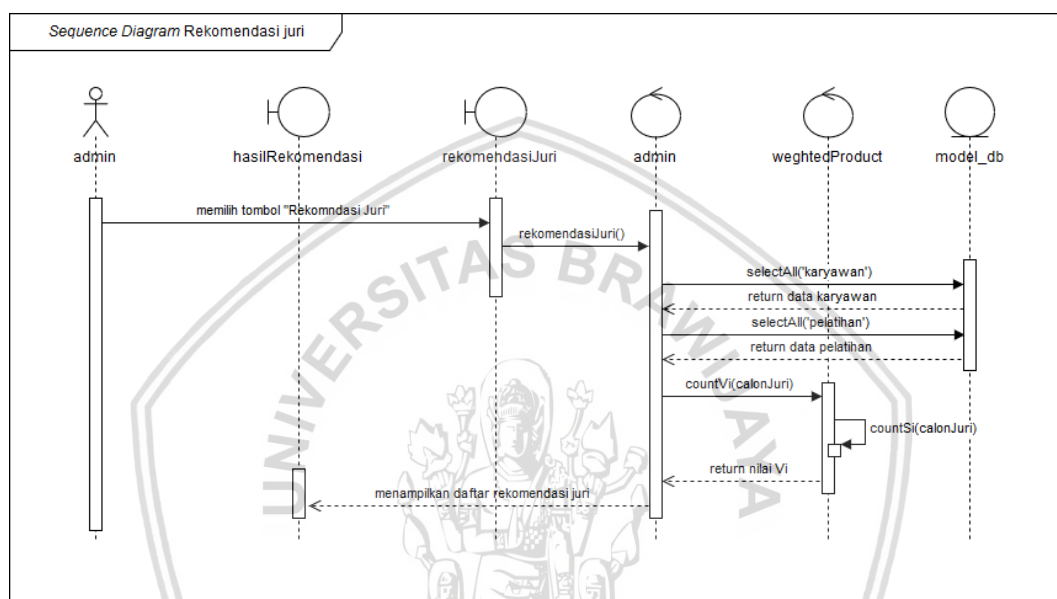


Gambar 5.2 Sequence diagram cek plagiasi.

Pada Gambar 5.2 menggambarkan *sequence diagram* cek plagiasi. Menjelaskan urutan proses dari *use case* cek plagiasi. Proses pertama adalah seorang tim telaah mengisi kolom abstrak pada halaman cekPlagiasi kemudian memilih tombol cek plagiasi maka sistem akan memanggil fungsi `cekPlagiasi()` pada kelas controller `timtelaah`. Pada fungsi `cekPlagiasi()` dilakukan proses *stemming* dengan memanggil fungsi `stemmer(text)` kemudian dilakukan proses tokenisasi dengan memanggil fungsi `tokenizer(text)`. Kemudian dilanjutkan dengan memanggil fungsi `get_data('abstrak', data1)` pada kelas model `model_db` untuk mendapatkan data latih abstrak dan akan melakukan return data latih ke kelas controller `timtelaah` yang selanjutnya data latih tersebut akan

dilakukan proses *stemming* dengan memanggil fungsi `stemmer(text)` dan dilakukan proses tokenisasi dengan memanggil fungsi `tokenizer(text)`. Setelah itu akan dipanggil method `tags_to_point(articles)` pada kelas `similarity` yang akan mengembalikan nilai berupa `tags`. Kemudian akan dilakukan proses perhitungan nilai *cosine similarity* dengan memanggil fungsi `cosine(a,b)` pada kelas `similarity` yang akan mengembalikan nilai berupa hasil persentase kemiripan abstrak ke kelas `controller` `timtelaah` yang kemudian akan dikirimkan ke halaman hasil.

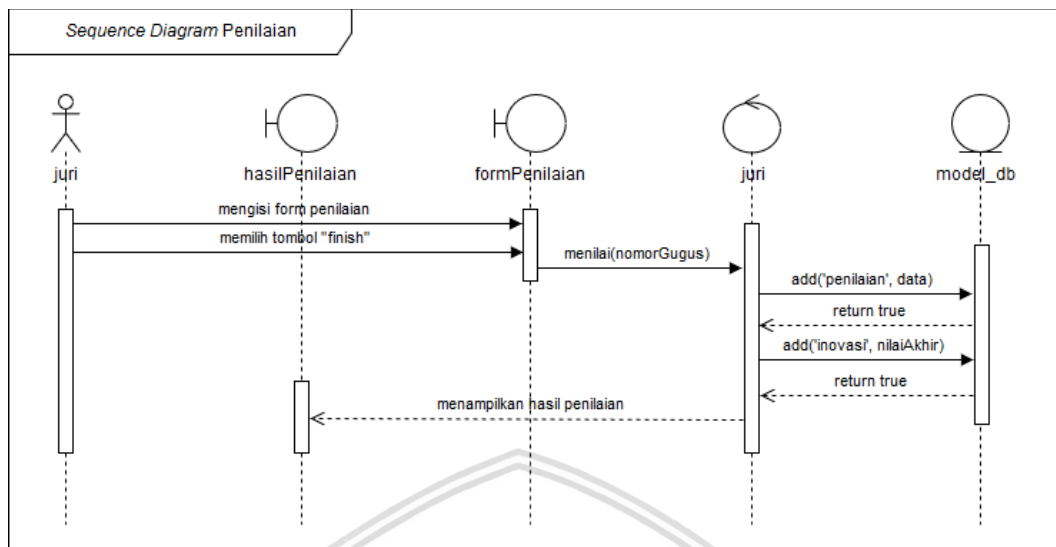
5.1.2.2 Sequence Diagram Rekomendasi Juri



Gambar 5.3 Sequence diagram Rekomendasi juri

Pada Gambar 5.3 menggambarkan *sequence diagram* rekomendasi juri. Menjelaskan urutan proses dari *use case* rekomendasi juri. Proses pertama adalah seorang admin memilih tombol "Rekomendasi Juri" pada halaman daftarCalonJuri kemudian sistem akan memanggil fungsi `rekomendasiJuri()` pada kelas *controller* admin, selanjutnya akan memanggil fungsi `selectAll('karyawan')` pada kelas *model* untuk mendapatkan data karyawan dan akan mengembalikan data karyawan ke kelas *controller* admin. selanjutnya akan memanggil fungsi `selectAll('pelatihan')` pada kelas *model* untuk mendapatkan data karyawan dan akan mengembalikan data pelatihan ke kelas *controller* admin. Kemudian data tersebut akan diolah pada fungsi `countVi(calonJuri)` di kelas *controller* `wegtedProduct` yang pada fungsi tersebut memanggil fungsi `countSi(calonJuri)` untuk proses algoritme *weighted Product* lalu akan mengembalikan nilai V_i sebagai hasil akhir penilaian untuk rekomendasi juri. Pada proses terakhir data rekomendasi juri yang akan ditampilkan pada halaman `hasilRekomendasi`.

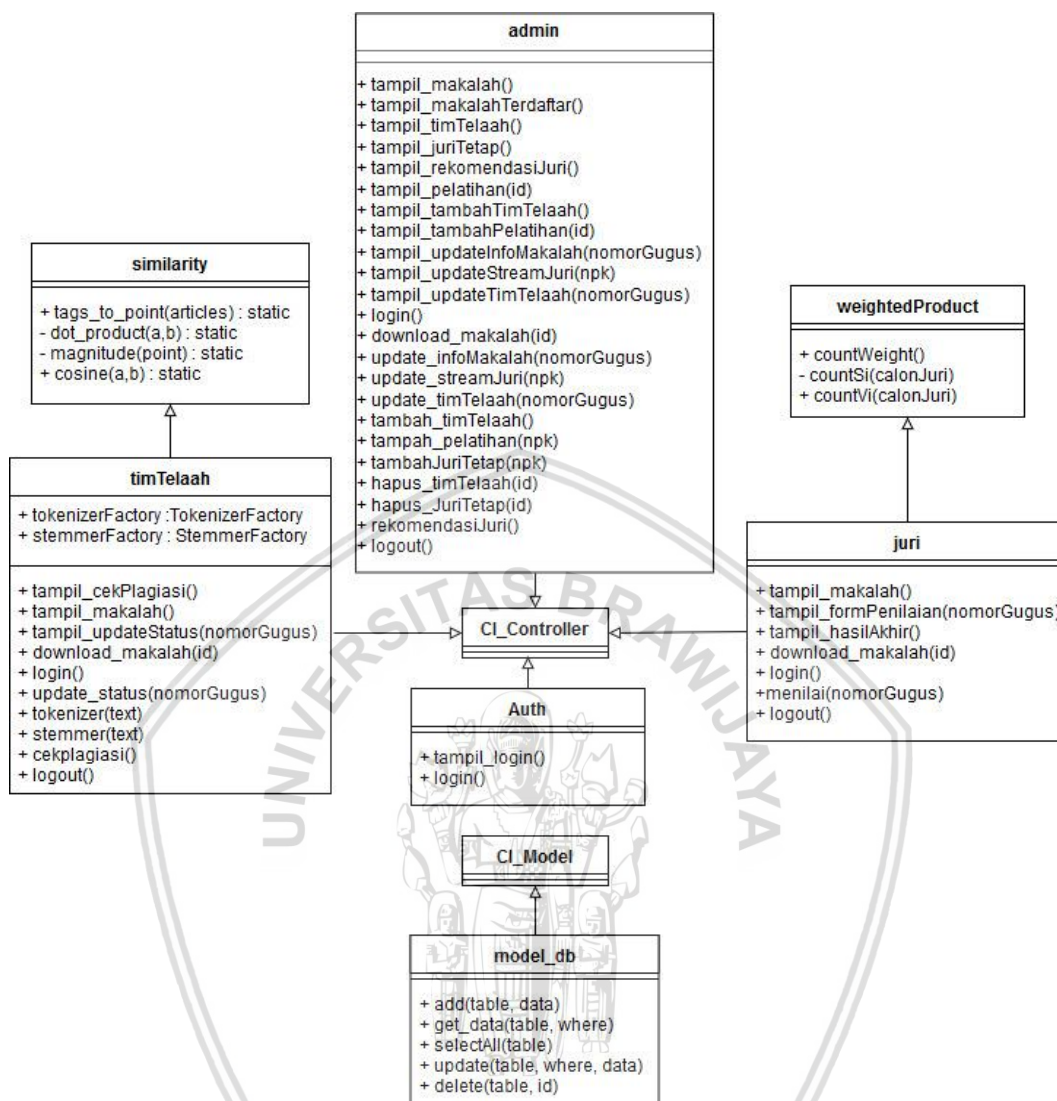
5.1.2.3 Sequence Diagram Menilai Makalah



Gambar 5.4 Sequence diagram Menilai Makalah

Pada Gambar 5.4 menggambarkan *sequence diagram* menilai makalah. Menjelaskan urutan proses dari *use case* menilai makalah. Proses pertama adalah seorang tim juri mengisi form penilaian pada halaman formPenilaian kemudian juri memilih tombol *finish*. Selanjutnya sistem akan mengolah nilai yang telah dimasukkan pada form penilaian pada fungsi `menilai()` yang terdapat pada kelas *controller* juri. Nilai tersebut kemudian akan disimpan dalam basis data sistem dengan memanggil fungsi `add(penilaian, data)` pada kelas *model* `model_db` yang akan mengembalikan nilai *true* ke kelas *controller* juri. Lalu selanjutnya dilakukan perhitungan nilai akhir pada kelas *controller* juri kemudian data akan disimpan dalam basis data sistem dengan memanggil fungsi `add(inovasi, data)` pada kelas *model* `model_db` yang akan mengembalikan nilai *true* ke kelas *controller* juri dan nilai akan ditampilkan pada halaman hasilPenilaian.

5.1.3 Class Diagram SIPO



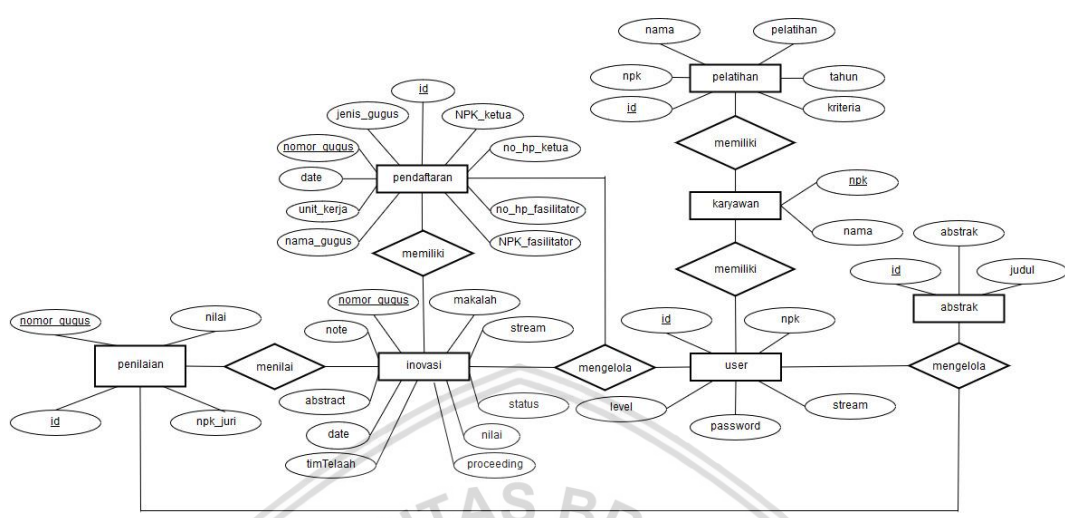
Gambar 5.5 Class diagram SIPO.

Gambar 5.5 merupakan *class diagram* dari sistem penjurian online menggunakan *framework Code Igniter*. *Class diagram* didapat dari *sequence diagram*. Terdapat control utama yaitu *CI_controller*, *admin*, *timtelaah*, *juri*, *auth*, *CI_model*, *model_db*. Control *auth*, *admin*, *timtelaah*, dan *juri* melakukan *inheritance* pada *CI_controller* yang berisi operasi-operasi yang dibutuhkan sistem untuk mengolah data-data yang dibutuhkan. Sedangkan *model_db* melakukan *inheritance* pada *CI_Model* yang berisi operasi-operasi untuk mengakses basis data.

5.1.4 Perancangan Data

Pada perancangan data digunakan *Entity relationship diagram* (ERD) dalam pemodelanya. ERD merupakan diagram yang menunjukkan hubungan antar entitas data sistem pada tahap perancangan data. pada gambar 5.5 merupakan ERD dari

requirement repository. Requirement repository merupakan basis data utama yang ada pada setiap sistem informasi.



Gambar 5.6 Entity Relationship Diagram SIPO.

Pada Gambar 5.6 ditunjukkan bahwa pada pengembangan sistem informasi penjuian online ini terdapat 9 entitas yaitu karyawan, user, pelatihan, pendaftaran, penilaian, inovasi, dan abstrak. entitas-entitas tersebut memiliki hubungan satu dengan yang lainnya untuk mempermudah pertukaran data secara dinamis.

5.1.5 Perancangan Algoritma

Pada sub bab perancangan algoritma ini akan dijelaskan algoritma yang digunakan untuk implementasi pembangunan sistem penjurian online menggunakan *framework code Igniter*.

Tabel 5.1 Pseudocode method CekPlagiasi pada kelas timtelaah

| | |
|----|--|
| 1 | Start |
| 2 | Text = input (abstrak) |
| 3 | stemms = hasil stemming text input |
| 4 | tokens = hasil tokenisasi text input |
| 5 | total = banyak data abstrak |
| 6 | jumlah = 0 |
| 7 | pembagi = 0 |
| 8 | for i to total Do |
| 9 | text1 = get abstrak ke i |
| 10 | stemms1 = hasil stemming text1 |
| 11 | tokens1 = hasil tokenisasi text1 |
| 12 | articles[i] = tokens1 |
| 13 | endfor |
| 14 | tags = memberi nilai pada setiap data tags |
| 15 | compare = memberi nilai setiap value array tokens berniai 1 dan |
| 16 | menampung value tags |
| 17 | for i to total Do |
| 18 | ak = memberi nilai setiap value array articles ke i dengan nilai |
| 19 | 1 dan menampung tags |
| 20 | hasil[i] = menyimpan data hasil proses perhitungan cosine |
| 21 | similarity data inputan dengan setiap data abstrak pada basis |
| 22 | data |
| 23 | jumlah += hasil[i] |

| | |
|----|------------------|
| 24 | if hasil[i] != 0 |
| 25 | pembagi++ |
| 26 | endif |
| 27 | endfor |
| 28 | end |

Pada Tabel 5.1 merupakan *pseudocode* method cekPlagiasi pada kelas timtelaah untuk melakukan proses cek plagiasi *internal*. Pada method ini diawali dengan menerima masukan berupa text dan kemudian masukan tersebut dilakukan *preprocessing text* berupa proses *stemming* dan tokenisasi. Selanjutnya dilakukan *preprocessing text* pada data-data abstrak yang digunakan sebagai data latih dari proses cek plagiasi. Kemudian dilakukan proses cosine untuk menentukan hasil persentase kecocokan kata dari text masukan dengan text pada data-data pada basis data.

Tabel 5.2 Pseudocode method rekomendasiJuri pada kelas admin

| | |
|----|---|
| 1 | Start |
| 2 | karyawan = data karyawan |
| 3 | pelatihan = data pelatihan |
| 4 | For i to jumlah karyawan Do |
| 5 | For j to jumlah pelatihan Do |
| 6 | If npk_karyawan[i] == npk_pelatihan[j] |
| 7 | If kriteria_pelatihan[j] == 1 |
| 8 | C1 = 1 |
| 9 | C2 = 1 |
| 10 | C3 = 1 |
| 11 | Foreach k to jumlah pelatihan Do |
| 12 | If npk_pelatihan[j] == npk_pelatihan[k] |
| 13 | If kriteria_pelatihan[k] == 2 |
| 14 | C1 = C1 + 1 |
| 15 | endif |
| 16 | else if kriteria_pelatihan[k] == 3 |
| 17 | C2 = C2 + 1 |
| 18 | endif |
| 19 | else if kriteria_pelatihan[k] == 4 |
| 20 | C3 = C3 + 1 |
| 21 | endif |
| 22 | endif |
| 23 | endif |
| 24 | endif |
| 25 | endif |
| 26 | endfor |
| 27 | BobotKriteria = [C1, C2, C3] |
| 28 | calonJuri = data calon juri |
| 30 | vi = hasil perhitungan dengan metode weighted product |
| 31 | end |

Pada Tabel 5.2 merupakan *pseudocode* method rekomendasiJuri pada kelas admin untuk melakukan proses rekomendasi juri dengan menggunakan algoritma *weighted product*. Pada method ini diawali dengan mengambil data karyawan serta data pelatihan pada basis data sistem. Selanjutnya dilakukan proses pembobotan pada setiap karyawan yang telah melakukan pelatihan juri pemula dan kemudian disimpan pada variabel bobotKriteria. Kemudian dilakukan proses *weighted product* dengan masukan data calon juri beserta bobot kriteria dari masing-masing juri yang didapat dari proses pembobotan sebelumnya.

Tabel 5.3 Pseudocode method penilaian pada kelas juri

| | |
|----|--|
| 1 | Start |
| 2 | nomorGugus = parameter |
| 3 | Nilai = input(nilai) |
| 4 | bobotSS_SSG = bobot nilai makalah jenis SS dan SSG |
| 5 | bobotFAME = bobot nilai makalah jenis FAME |
| 6 | bobotPKM_GKM = bobot nilai makalah jenis PKM dan GKM |
| 7 | jumlah = 0 |
| 8 | hasil = 0 |
| 9 | hasilAkhir = 0 |
| 10 | Pendaftaran = data pendaftaran |
| 11 | If (jenis gugus pendaftaran = SSG atau SS) |
| 12 | for i in jumlah nilai Do |
| 13 | Hasil = hasil + (nilai[i]*bobotSS_SSG[i]) |
| 14 | endfor |
| 15 | endif |
| 16 | else if (jenis gugus pendaftaran = FAME) |
| 17 | for i in jumlah nilai Do |
| 18 | Hasil = hasil + (nilai[i]*bobotFAME[i]) |
| 19 | endfor |
| 20 | endif |
| 21 | else if (jenis gugus pendaftaran = PKM atau GKM) |
| 22 | for i in jumlah nilai Do |
| 23 | Hasil = hasil + (nilai[i]*bobotPKM_GKM[i]) |
| 24 | endfor |
| 25 | endif |
| 26 | Penilaian = data penilaian |
| 27 | for i in jumlah penilaian Do |
| 28 | if nomor gugus pada data penilaian = nomorGugus |
| 29 | Jumlah = jumlah+1 |
| 30 | hasilAkhir = hasilAkhir + nilai pada data penilaian |
| 31 | endif |
| 32 | endfor |
| 33 | hasilAkhir = hasilAkhir/jumlah |
| 34 | end |

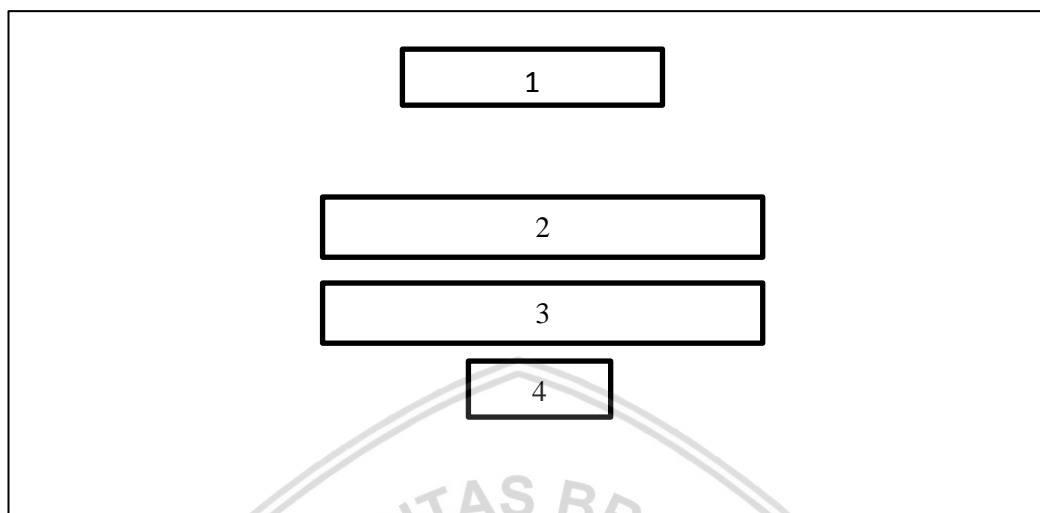
Pada Tabel 5.3 merupakan *pseudocode* method penilaian pada kelas juri untuk melakukan proses penilaian makalah dengan menghitung hasil penilaian setiap kriteria yang diberikan oleh juri. Pada method ini diawali dengan mengambil data masukan juri berupa nilai yang diberikan juri pada form penilaian. Kemudian dilakukan perhitungan sesuai dengan jenis makalah yang sedang dinilai dan menyimpan nilai tersebut pada basis data sistem serta dilakukan juga perhitungan nilai akhir yang mana perhitungan tersebut adalah perhitungan rata-rata jumlah seluruh nilai yang diberikan oleh juri yang telah menilai sebuah makalah.

5.1.6 Perancangan Antarmuka

Pada perancangan antarmuka terdapat beberapa sampel rancangan antarmuka sistem seperti halaman utama, halaman login, halaman home, halaman cek plagiasi, halaman rekomendasi sistem, dll. Perancangan antarmuka pada subab ini akan dijadikan sebagai acuan dalam implementasi antarmuka sistem.

5.1.6.1 Perancangan Antarmuka Halaman Login

Perancangan antarmuka halaman login digambarkan pada Gambar 5.7, dan penjelasan setiap objek pada perancangan antarmuka dijelaskan pada tabel 5.4.



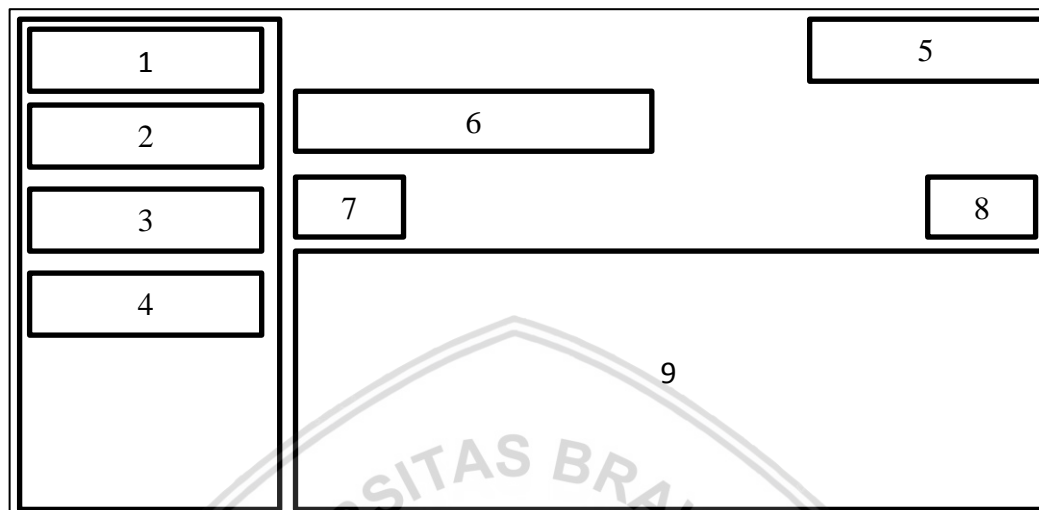
Gambar 5.7 Perancangan Antarmuka Halaman Login

Tabel 5.4 penjelasan Perancangan Antarmuka Halaman Login

| No. | Nama Objek | Tipe | Keterangan |
|-----|---------------------|---------------|-------------------------------------|
| 1. | Logo Sistem | Logo | Menunjukkan logo sistem |
| 2. | Kolom npk | Text | Kolom untuk masukan npk |
| 3. | Kolom password | Text | Kolom untuk masukan password |
| 4. | <i>Button</i> login | <i>Button</i> | <i>Button</i> untuk melakukan login |

5.1.6.2 Perancangan Antarmuka Halaman Home Tim Telaah

Perancangan antarmuka halaman home tim telaah digambarkan pada Gambar 5.8, dan penjelasan setiap objek pada perancangan antarmuka dijelaskan pada tabel 5.5.



Gambar 5.8 Perancangan Antarmuka Halaman Home Tim Telaah

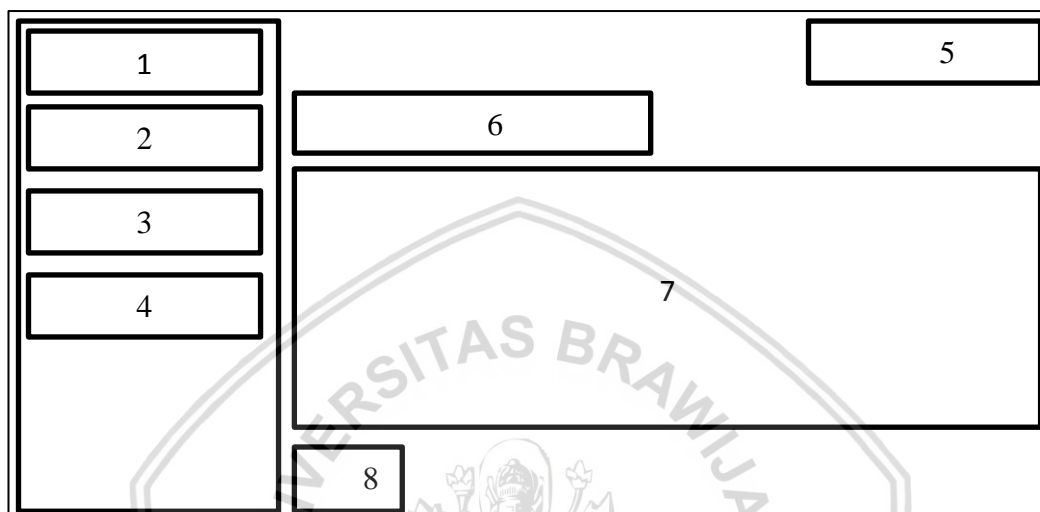
Tabel 5.5 Penjelasan Perancangan Antarmuka Halaman Home Tim Telaah

| No. | Nama Objek | Tipe | Keterangan |
|-----|---|-----------------|--|
| 1. | Logo Actor | Logo | Menunjukkan logo Actor |
| 2. | Logo text nama Actor | Logo text | Menunjukkan nama Actor yang sedang menggunakan sistem |
| 3. | Navigation text makalah | Navigation text | untuk menampilkan halaman data makalah |
| 4. | Navigation text cek plagiasi | Navigation text | Untuk menampilkan form cek plagiasi |
| 5. | Logo text nama Actor | Logo text | Untuk menampilkan button logout |
| 6. | Logo text keterangan halaman data makalah | Logo text | Menunjukkan keterangan halaman data makalah |
| 7. | Kolom jumlah kolom tabel | Int | Menentukan jumlah kolom yang akan ditampilkan pada tabel |
| 8. | Kolom search | Text | Kolom masukan text yang ingin ditemukan pada tabel. |

| | | | |
|----|---------------|-------|--|
| 9. | Tabel makalah | Tabel | Menampilkan data makalah yang akan dikaji. |
|----|---------------|-------|--|

5.1.6.3 Perancangan Antarmuka Halaman Cek Plagiasi

Perancangan antarmuka halaman cek plagiasi digambarkan pada Gambar 5.9, dan penjelasan setiap objek pada perancangan antarmuka dijelaskan pada tabel 5.6.



Gambar 5.9 Perancangan Antarmuka Halaman Cek Plagiasi

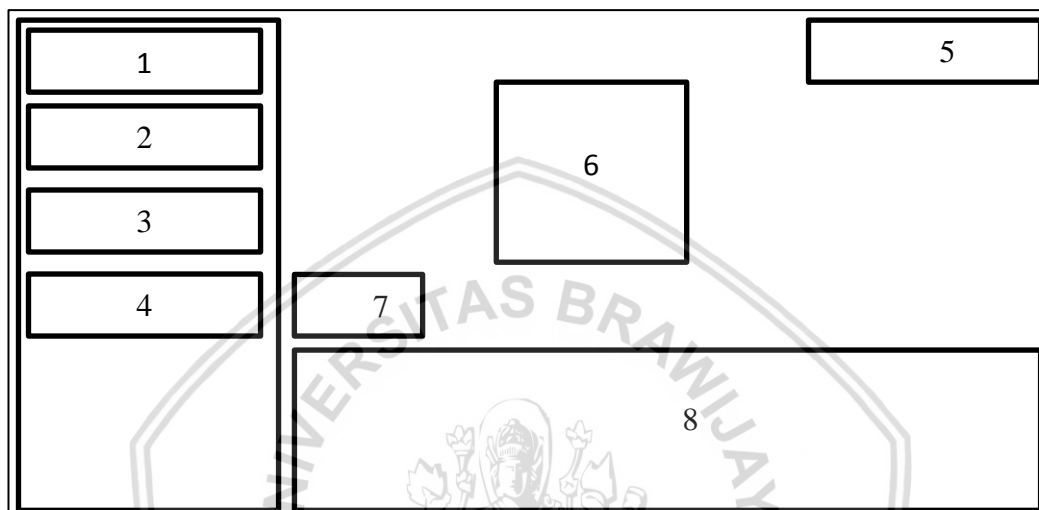
Tabel 5.6 Penjelasan Perancangan Antarmuka Halaman Cek Plagiasi

| No. | Nama Objek | Tipe | Keterangan |
|-----|---|-----------------|---|
| 1. | Logo Actor | Logo | Menunjukkan logo Actor |
| 2. | Logo text nama Actor | Logo text | Menunjukkan nama Actor yang sedang menggunakan sistem |
| 3. | Navigation text makalah | Navigation text | untuk menampilkan halaman data makalah |
| 4. | Navigation text cek plagiasi | Navigation text | Untuk menampilkan form cek plagiasi |
| 5. | Logo text nama Actor | Logo text | Untuk menampilkan button logout |
| 6. | Logo text keterangan form masukan abstrak | Logo text | Menunjukkan keterangan form masukan abstrak |
| 7. | Text area abstrak | Text | Untuk masukan text abstrak |

| | | | |
|----|---------------------|--------|---|
| 8. | Button cek plagiasi | Button | Button untuk menampilkan halaman hasil cek plagiasi |
|----|---------------------|--------|---|

5.1.6.4 Perancangan Antarmuka Halaman Hasil Cek Plagiasi

Perancangan antarmuka halaman hasil cek plagiasi digambarkan pada Gambar 5.10, dan penjelasan setiap objek pada perancangan antarmuka dijelaskan pada tabel 5.7.



Gambar 5.10 Perancangan Antarmuka Halaman Hasil Cek Plagiasi

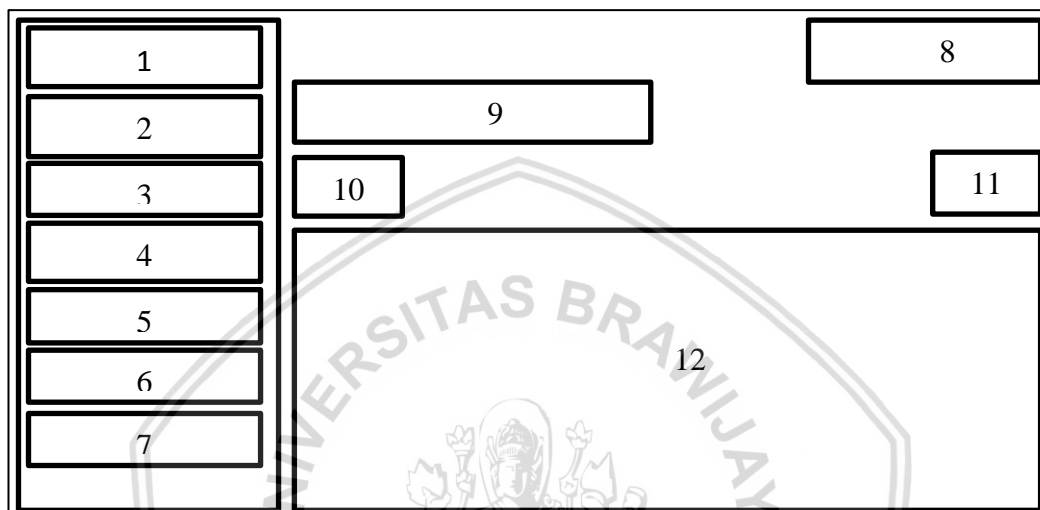
Tabel 5.7 Penjelasan Perancangan Antarmuka Halaman Hasil Cek Plagiasi

| No. | Nama Objek | Tipe | Keterangan |
|-----|---|-----------------|---|
| 1. | Logo Actor | Logo | Menunjukkan logo Actor |
| 2. | Logo text nama Actor | Logo text | Menunjukkan nama Actor yang sedang menggunakan sistem |
| 3. | Navigation text makalah | Navigation text | untuk menampilkan halaman data makalah |
| 4. | Navigation text cek plagiasi | Navigation text | Untuk menampilkan form cek plagiasi |
| 5. | Logo text nama Actor | Logo text | Untuk menampilkan button logout |
| 6. | Diagram pie hasil cek plagiasi | Diagram pie | Menampilkan persentase hasil plagiasi dengan bentuk diagram pie |
| 7. | Logo text keterangan hasil cek plagiasi | Logo text | Menampilkan logo text keterangan hasil cek plagiasi |

| | | | |
|----|--------------------------|-------|---|
| 8. | Tabel hasil cek plagiasi | Tabel | Menampilkan data persentase plagiasi setiap abstrak pada basis data |
|----|--------------------------|-------|---|

5.1.6.5 Perancangan Antarmuka Halaman Home Admin

Perancangan antarmuka halaman home admin digambarkan pada Gambar 5.11, dan penjelasan setiap objek pada perancangan antarmuka dijelaskan pada tabel 5.8.



Gambar 5.11 Perancangan Antarmuka Halaman Home Admin

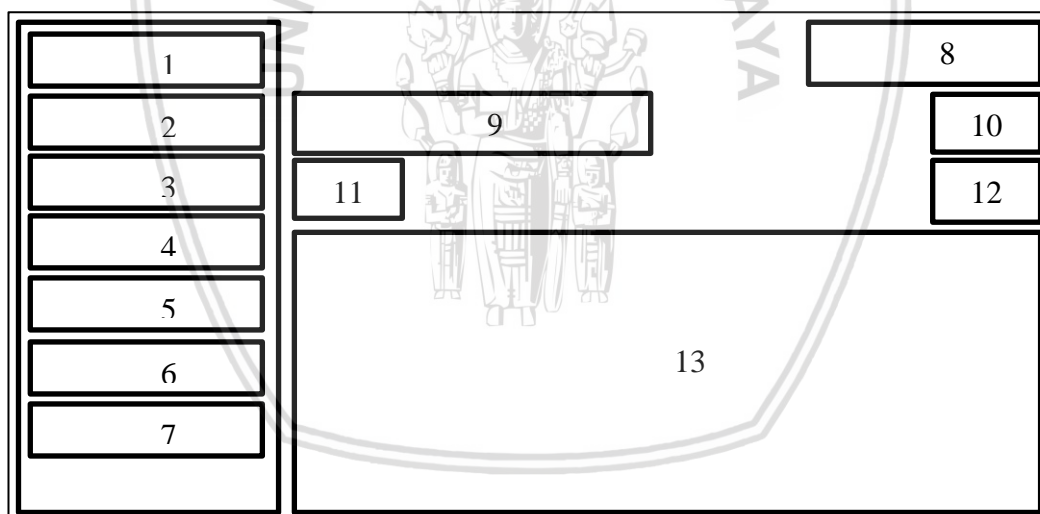
Tabel 5.8 Penjelasan Perancangan Antarmuka Halaman Home Admin

| No. | Nama Objek | Tipe | Keterangan |
|-----|-----------------------------------|-----------------|---|
| 1. | Logo Actor | Logo | Menunjukkan logo Actor |
| 2. | Logo text nama Actor | Logo text | Menunjukkan nama Actor yang sedang menggunakan sistem |
| 3. | Navigation text makalah | Navigation text | untuk menampilkan halaman data makalah |
| 4. | Navigation text makalah terdaftar | Navigation text | Untuk menampilkan halaman data makalah terdaftar |
| 5. | Navigation text tim telaah | Navigation text | Untuk menampilkan halaman data tim telaah |
| 6. | Navigation text rekomendasi juri | Navigation text | Untuk menampilkan halaman calon juri |
| 7. | Navigation text juri | Navigation text | Untuk menampilkan halaman juri tetap |

| | | | |
|-----|--|-----------|--|
| 8. | Logo text nama <i>Actor</i> | Logo text | Untuk menampilkan <i>button</i> logout |
| 9. | Logo keterangan halaman data makalah lolos | Logo text | Menunjukkan keterangan halaman data makalah lolos |
| 10. | Kolom jumlah kolom tabel | Int | Menentukan jumlah kolom yang akan ditampilkan pada tabel |
| 11. | Kolom search | Text | Kolom masukan text yang ingin ditemukan pada tabel. |
| 12. | Tabel makalah | Tabel | Menampilkan data makalah yang akan dikaji. |

5.1.6.6 Perancangan Antarmuka Halaman Daftar Tim Telaah

Perancangan antarmuka halaman daftar tim telaah digambarkan pada Gambar 5.12, dan penjelasan setiap objek pada perancangan antarmuka dijelaskan pada tabel 5.9.



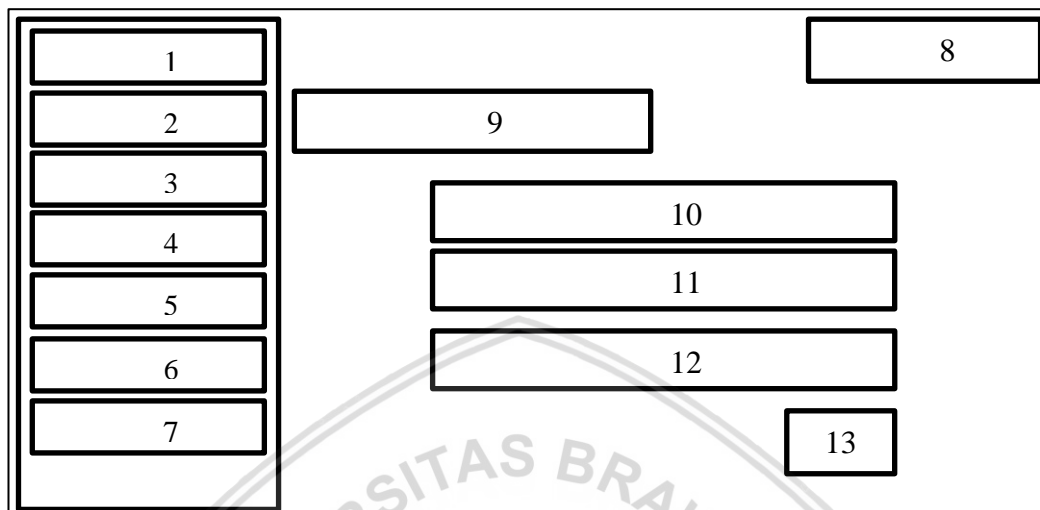
Gambar 5.12 Perancangan Antarmuka Halaman Daftar Tim Telaah

Tabel 5.9 Penjelasan Perancangan Antarmuka Halaman Daftar Tim Telaah

| No. | Nama Objek | Tipe | Keterangan |
|-----|--|------------------------|--|
| 1. | Logo <i>Actor</i> | Logo | Menunjukkan logo <i>Actor</i> |
| 2. | Logo text nama <i>Actor</i> | Logo text | Menunjukkan nama <i>Actor</i> yang sedang menggunakan sistem |
| 3. | <i>Navigation text</i> makalah | <i>Navigation text</i> | untuk menampilkan halaman data makalah |
| 4. | <i>Navigation text</i> makalah terdaftar | <i>Navigation text</i> | Untuk menampilkan halaman data makalah terdaftar |
| 5. | <i>Navigation text</i> tim telaah | <i>Navigation text</i> | Untuk menampilkan halaman data tim telaah |
| 6. | <i>Navigation text</i> rekomendasi juri | <i>Navigation text</i> | Untuk menampilkan halaman calon juri |
| 7. | <i>Navigation text</i> juri | <i>Navigation text</i> | Untuk menampilkan halaman juri tetap |
| 8. | Logo text nama <i>Actor</i> | Logo text | Untuk menampilkan <i>button</i> logout |
| 9. | Logo text keterangan halaman data tim telaah | Logo text | Menunjukkan keterangan halaman data tim telaah |
| 10. | <i>Button</i> tambah tim telaah | <i>Button</i> | <i>Button</i> untuk menampilkan halaman tambah tim telaah |
| 11. | Kolom jumlah kolom tabel | Int | Menentukan jumlah kolom yang akan ditampilkan pada tabel |
| 12. | Kolom search | Text | Kolom masukan text yang ingin ditemukan pada tabel |
| 13. | Tabel tim telaah | Tabel | Menampilkan data tim telaah |

5.1.6.7 Perancangan Antarmuka Halaman Tambah Tim Telaah

Perancangan antarmuka halaman tambah tim telaah digambarkan pada Gambar 5.13, dan penjelasan setiap objek pada perancangan antarmuka dijelaskan pada tabel 5.10.



Gambar 5.13 Perancangan Antarmuka Halaman Tambah Tim Telaah

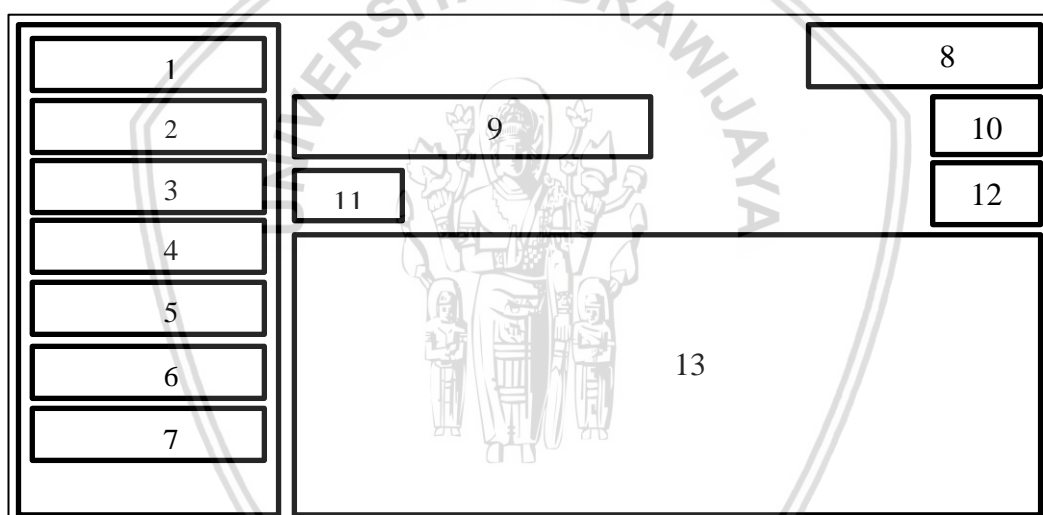
Tabel 5.10 Penjelasan Perancangan Antarmuka Halaman Tambah Tim Telaah

| No. | Nama Objek | Tipe | Keterangan |
|-----|---|-----------------|---|
| 1. | Logo Actor | Logo | Menunjukkan logo Actor |
| 2. | Logo text nama Actor | Logo text | Menunjukkan nama Actor yang sedang menggunakan sistem |
| 3. | Navigation text makalah | Navigation text | untuk menampilkan halaman data makalah |
| 4. | Navigation text makalah terdaftar | Navigation text | Untuk menampilkan halaman data makalah terdaftar |
| 5. | Navigation text tim telaah | Navigation text | Untuk menampilkan halaman data tim telaah |
| 6. | Navigation text rekomendasi juri | Navigation text | Untuk menampilkan halaman calon juri |
| 7. | Navigation text juri tetap | Navigation text | Untuk menampilkan halaman juri tetap |
| 8. | Logo text nama Actor | Logo text | Untuk menampilkan button logout |
| 9. | Logo text keterangan form tambah tim telaah | Logo text | Menunjukkan keterangan form tambah tim telaah |

| | | | |
|-----|----------------|--------|---|
| 10. | Kolom npk | Text | Kolom untuk masukan npk tim telaah |
| 11. | Kolom nama | Text | Kolom untuk masukan nama tim telaah |
| 12. | Kolom password | Text | Kolom untuk masukan password tim telaah |
| 13. | Button tambah | Button | Button untuk menambahkan data tim telaah yang dimasukkan pada form tambah tim telaah. |

5.1.6.8 Perancangan Antarmuka Halaman Data Calon Juri

Perancangan antarmuka halaman data calon juri digambarkan pada Gambar 5.14, dan penjelasan setiap objek pada perancangan antarmuka dijelaskan pada tabel 5.11.



Gambar 5.14 Perancangan Antarmuka Halaman Data Calon Juri

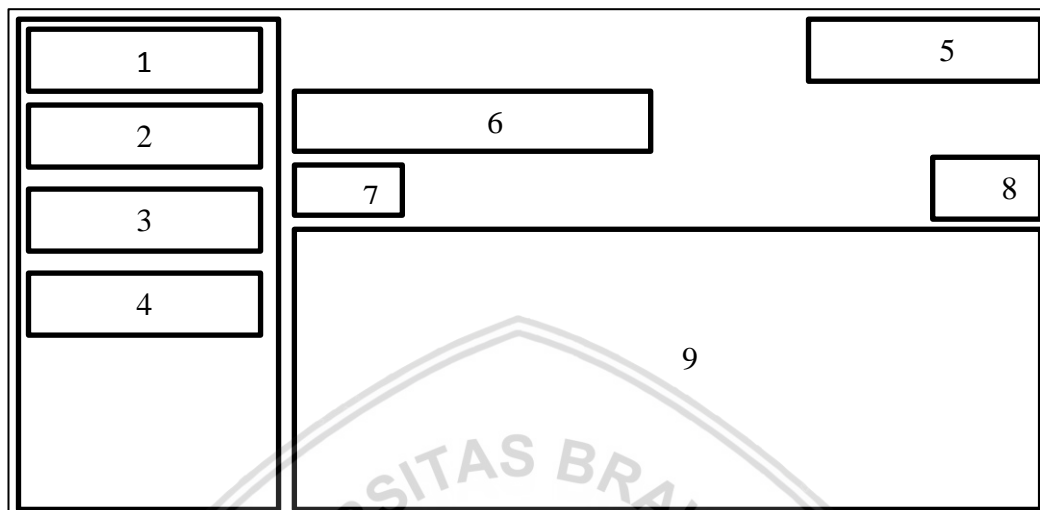
Tabel 5.11 Penjelasan Perancangan Antarmuka Halaman Data Calon Juri

| No. | Nama Objek | Tipe | Keterangan |
|-----|-----------------------------------|-----------------|---|
| 1. | Logo Actor | Logo | Menunjukkan logo Actor |
| 2. | Logo text nama Actor | Logo text | Menunjukkan nama Actor yang sedang menggunakan sistem |
| 3. | Navigation text makalah | Navigation text | untuk menampilkan halaman data makalah |
| 4. | Navigation text makalah terdaftar | Navigation text | Untuk menampilkan halaman data makalah terdaftar |

| | | | |
|-----|--|------------------------|--|
| 5. | <i>Navigation text</i> tim telaah | <i>Navigation text</i> | Untuk menampilkan halaman data tim telaah |
| 6. | <i>Navigation text</i> rekomendasi juri | <i>Navigation text</i> | Untuk menampilkan halaman calon juri |
| 7. | <i>Navigation text</i> juri | <i>Navigation text</i> | Untuk menampilkan halaman juri tetap |
| 8. | Logo text nama <i>Actor</i> | Logo text | Untuk menampilkan <i>button</i> logout |
| 9. | Logo text keterangan halaman data calon juri | Logo text | Menunjukkan keterangan halaman data calon juri |
| 10. | <i>Button</i> rekomendasi juri | <i>Button</i> | <i>Button</i> untuk menampilkan halaman hasil rekomendasi juri |
| 11. | Kolom jumlah kolom tabel | Int | Menentukan jumlah kolom yang akan ditampilkan pada tabel |
| 12. | Kolom search | Text | Kolom masukan text yang ingin ditemukan pada tabel |
| 13. | Tabel calon juri | Tabel | Menampilkan data calon juri |

5.1.6.9 Perancangan Antarmuka Halaman Home Juri

Perancangan antarmuka halaman home juri digambarkan pada Gambar 5.15, dan penjelasan setiap objek pada perancangan antarmuka dijelaskan pada tabel 5.12.



Gambar 5.15 Perancangan Antarmuka Halaman Home Juri

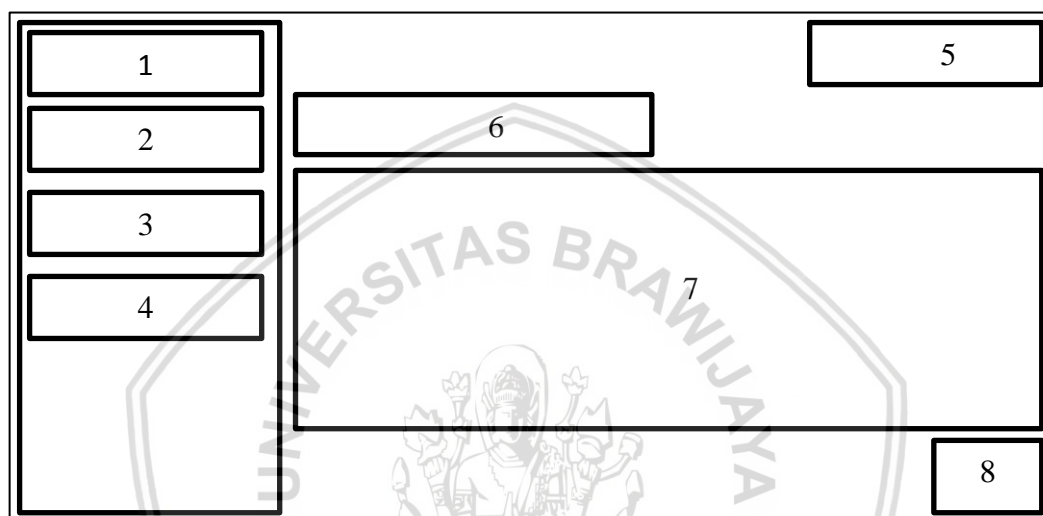
Tabel 5.12 Penjelasan Perancangan Antarmuka Halaman Home Juri

| No. | Nama Objek | Tipe | Keterangan |
|-----|--------------------------------------|-----------------|---|
| 1. | Logo Actor | Logo | Menunjukkan logo Actor |
| 2. | Logo text nama Actor | Logo text | Menunjukkan nama Actor yang sedang menggunakan sistem |
| 3. | Navigation text makalah | Navigation text | untuk menampilkan halaman data makalah |
| 4. | Navigation text hasil akhir | Navigation text | Untuk menampilkan halaman data makalah dan hasil akhir penilaian |
| 5. | Logo text nama Actor | Logo text | Untuk menampilkan button logout |
| 6. | Logo text keterangan halaman makalah | Logo text | Menunjukkan keterangan halaman data makalah dengan stream yang sesuai dengan stream juri yang sedang mengakses sistem |
| 7. | Kolom jumlah kolom tabel | Int | Menentukan jumlah kolom yang akan ditampilkan pada tabel |

| | | | |
|----|---------------|-------|---|
| 8. | Kolom search | Text | Kolom masukan text yang ingin ditemukan pada tabel. |
| 9. | Tabel makalah | Tabel | Menampilkan data makalah yang lolos. |

5.1.6.10 Perancangan Antarmuka Halaman *Form* Penilaian Juri

Perancangan antarmuka halaman *form* penilaian juri digambarkan pada Gambar 5.16, dan penjelasan setiap objek pada perancangan antarmuka dijelaskan pada tabel 5.13.



Gambar 5.16 Perancangan Antarmuka Halaman *Form* Penilaian Juri

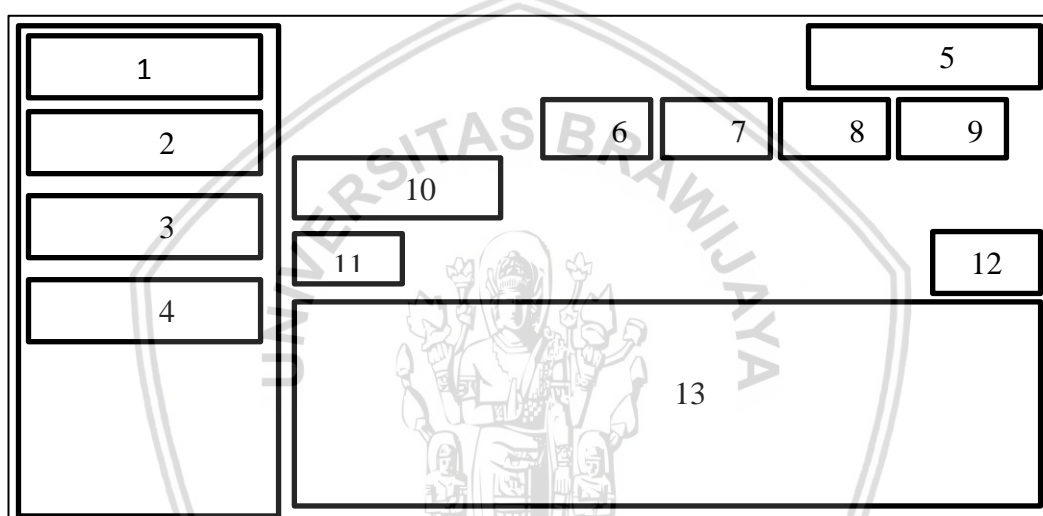
Tabel 5.13 Penjelasan Perancangan Antarmuka Halaman *Form* Penilaian Juri

| No. | Nama Objek | Tipe | Keterangan |
|-----|-------------------------------------|-----------------|---|
| 1. | Logo Actor | Logo | Menunjukkan logo Actor |
| 2. | Logo text nama Actor | Logo text | Menunjukkan nama Actor yang sedang menggunakan sistem |
| 3. | Navigation text makalah | Navigation text | untuk menampilkan halaman data makalah |
| 4. | Navigation text hasil akhir | Navigation text | Untuk menampilkan halaman data makalah dan hasil akhir penilaian |
| 5. | Logo text nama Actor | Logo text | Untuk menampilkan <i>button</i> logout |
| 6. | Logo text keterangan form penilaian | Logo text | Menunjukkan keterangan form penilaian sesuai engan jenis gugus dan judul makalah yang dipilih |

| | | | |
|----|-----------------------|---------------|---|
| 7. | <i>Form</i> penilaian | <i>form</i> | Menunjukkan <i>form</i> penilaian dengan kriteria penilaian yang harus diisi dengan memilih angka 1 sampai 10 dengan bentuk <i>radio button</i> |
| 8. | <i>Button</i> nilai | <i>Button</i> | <i>Button</i> untuk menampilkan jumlah nilai yang diberikan. |

5.1.6.11 Perancangan Antarmuka Halaman Hasil Akhir Penilaian

Perancangan antarmuka halaman hasil akhir penilaian digambarkan pada Gambar 5.17, dan penjelasan setiap objek pada perancangan antarmuka dijelaskan pada tabel 5.14.



Gambar 5.17 Perancangan Antarmuka Halaman Hasil Akhir Penilaian

Tabel 5.14 Penjelasan Perancangan Antarmuka Halaman Hasil Akhir Penilaian

| No. | Nama Objek | Tipe | Keterangan |
|-----|------------------------------------|------------------------|--|
| 1. | Logo <i>Actor</i> | Logo | Menunjukkan logo <i>Actor</i> |
| 2. | Logo text nama <i>Actor</i> | Logo text | Menunjukkan nama <i>Actor</i> yang sedang menggunakan sistem |
| 3. | <i>Navigation text</i> makalah | <i>Navigation text</i> | untuk menampilkan halaman data makalah |
| 4. | <i>Navigation text</i> hasil akhir | <i>Navigation text</i> | Untuk menampilkan halaman data makalah dan hasil akhir penilaian |
| 5. | Logo text nama <i>Actor</i> | Logo text | Untuk menampilkan <i>button</i> logout |

| | | | |
|-----|---------------------------------------|-----------|---|
| 6. | Tab bar <i>stream</i> 4 | Tab bar | Untuk meampilkan data nilai akhir makalah <i>stream</i> 4 |
| 7. | Tab bar <i>stream</i> 3 | Tab bar | Untuk meampilkan data nilai akhir makalah <i>stream</i> 3 |
| 8. | Tab bar <i>stream</i> 2 | Tab bar | Untuk meampilkan data nilai akhir makalah <i>stream</i> 2 |
| 9. | Tab bar <i>stream</i> 1 | Tab bar | Untuk meampilkan data nilai akhir makalah <i>stream</i> 1 |
| 10. | Logo text keterangan <i>stream</i> | Logo text | Menunjukkan keterangan <i>stream</i> makalah |
| 11. | Kolom jumlah kolom tabel | Int | Menentukan jumlah kolom yang akan ditampilkan pada tabel |
| 12. | Kolom search | Text | Kolom masukan text yang ingin ditemukan pada tabel. |
| 13. | Tabel makalah | Tabel | Menampilkan data makalah sesuai <i>stream</i> |

5.2 Implementasi

Pada sub bab ini akan menjelaskan proses implementasi sistem penjurian online menggunakan *framework Code Igniter*. Terdapat spesifikasi pengembangan sistem, implementasi basis data, implementasi laman sistem, serta implementasi kode program.

5.2.1 Spesifikasi Pengembangan Sistem

Pada sub bab spesifikasi pengembangan sistem ini terdapat 2 bagian yang akan dijelaskan yaitu: spesifikasi perangkat keras dan spesifikasi perangkat lunak. Spesifikasi ini adalah spesifikasi yang digunakan dalam melakukan pengembangan sistem penjurian menggunakan *framework Code Igniter* dan library sastrawi pada proses *stemming* dan tokenisasi untuk *preprocessing text* dalam fungsi cek plagiasi.

5.2.1.1 Spesifikasi perangkat keras

Tabel 5.16 merupakan spesifikasi perangkat keras yang digunakan dalam melakukan implementasi pengembangan sistem penjurian online.

Tabel 5.15 Spesifikasi perangkat keras

| Nama Komponen | Spesifikasi |
|------------------------|---------------|
| <i>Processor</i> | Intel Core i7 |
| <i>Hard Disk Drive</i> | 256 GB |

| | |
|---------------------|------------|
| RAM | 8 GB |
| <i>System Model</i> | Asus X450L |

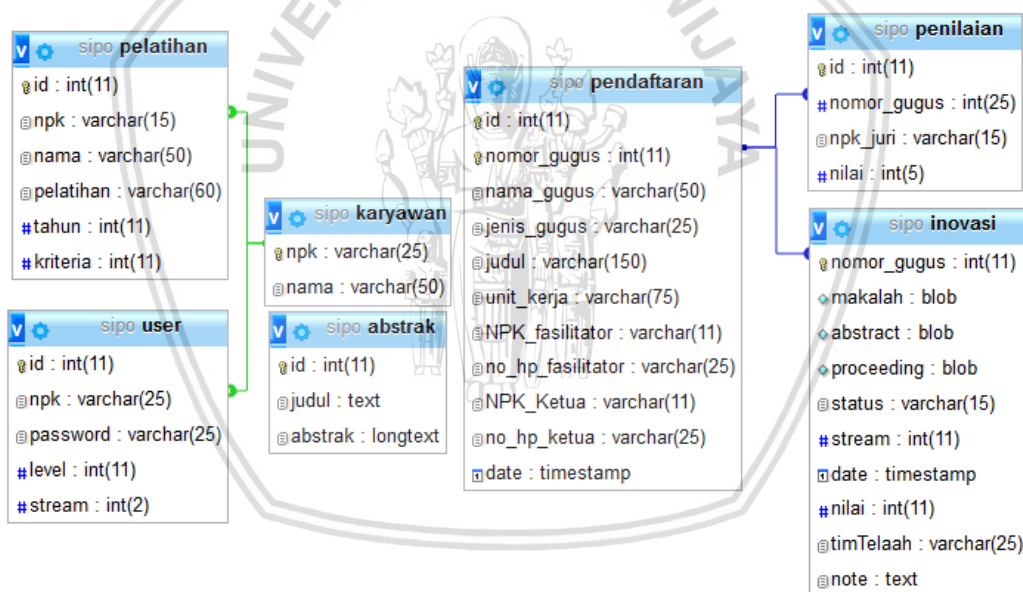
5.2.1.2 Spesifikasi Perangkat Lunak

Tabel 5.17 merupakan spesifikasi perangkat lunak yang digunakan dalam melakukan implementasi pengembangan sistem penjurian online.

Tabel 5.16 Spesifikasi perangkat lunak

| Nama Komponen | Spesifikasi |
|-----------------------------|--------------------|
| <i>Operating System</i> | Windows 8.1 64-bit |
| <i>Programming Language</i> | PHP |
| <i>Text Editor</i> | Notepad++ |
| <i>Server</i> | Apache |

5.2.2 Implementasi Basis data



Gambar 5.18 Physical Data Model basis data requirement repository.

Gambar 5.19 merupakan *Physical Data Model* basis data *requirement repository*. Pada basis data ini terdapat 7 tabel yang digunakan untuk menyimpan data-data yang dibutuhkan sistem. Tabel tersebut yaitu tabel karyawan, user, pendaftaran, inovasi, pelatihan, penilaian serta tabel abstrak. tabel-tabel tersebut memiliki hubungan serta kardinalitas antara tabel satu dengan tabel yang lainnya. *Physical Data Model* merupakan hasil normalisasi dari ERD yang telah didefinisikan pada sub bab 5.1.3.

5.2.3 Implementasi Kode Program

5.2.3.1 Implementasi Kode Program Kelas timTelaah

Nama Fungsi : cekPlagiasi()

Source Code :

Kode Program 5.1 Kode Program fungsi cekPlagiasi pada kelas timTelaah

| | |
|----|---|
| 1 | public function cekPlagiasi(){ |
| 2 | \$data['timtelaah']=\$this->model_db-> |
| 3 | selectAll('timtelaah'); |
| 4 | \$data['abstrak']=\$this->model_db-> |
| 5 | selectAll('abstrak'); |
| 6 | \$text = \$this->input->post('abstrak'); |
| 7 | \$stemms = \$this->CI->stemmer(\$text); |
| 8 | \$tokens = \$this->CI->tokenizer(\$stemms); |
| 9 | \$total = \$this->db->count_all_results('abstrak'); |
| 10 | \$jumlah=0; |
| 11 | \$pembagi=0; |
| 12 | for(\$i=0; \$i<\$total; \$i++){ |
| 13 | \$datal['id'] = (\$i+1); |
| 14 | \$abstrak=\$this->model_db-> |
| 15 | get_data('abstrak',\$datal)->row(); |
| 16 | \$text1 = \$abstrak->abstrak; |
| 17 | \$stemms1 = \$this->CI->stemmer(\$text1); |
| 18 | \$tokens1 = \$this->CI->tokenizer(\$stemms1); |
| 19 | \$articles[\$i] = array(|
| 20 | "article" => \$abstrak->judul, |
| 21 | "tags" => \$tokens1 |
| 22 |); |
| 23 | \$data['articles'][\$i] = (|
| 24 | \$articles[\$i]['article'] |
| 25 |); |
| 26 | } |
| 27 | \$tags = Similarity::tags_to_point(\$articles); |
| 28 | \$compare = array_fill_keys(\$tokens, 1) + \$tags; |
| 29 | for(\$i=0; \$i<\$total; \$i++){ |
| 30 | \$ak = array_fill_keys(\$articles[\$i]['tags'], |
| 31 | 1) + \$tags; |
| 32 | \$hasil[\$i]=(Similarity::cosine(\$compare, |
| 33 | \$ak))*100; |
| 34 | \$jumlah += \$hasil[\$i]; |
| 35 | if(\$hasil[\$i] != 0){ |
| 36 | \$pembagi++; |
| 37 | } |
| 38 | \$data['score'][\$i] = (|
| 39 | \$hasil[\$i] |
| 40 |); |
| 41 | } |
| 42 | \$data['hasil'] = \$jumlah/\$pembagi; |
| 43 | \$this->load->view('hasilCekPlagiasi',\$data); |
| 44 | } |

Penjelasan :

Tabel 5.17 Penjelasan kode program fungsi cekPlagiasi pada kelas timTelaah

| | |
|-----|---|
| 1 | Deklarasi fungsi cekPlagiasi |
| 2-3 | Inisialisasi variabel untuk menampung data tim telaah |
| 4-5 | Inisialisasi variabel untuk menampung data abstrak |
| 6 | Inisialisasi variabel untuk menampung input text |

| | |
|-------|---|
| 7 | Inisialisasi variabel untuk menampung hasil <i>stemming</i> input text |
| 8 | Inisialisasi variabel untuk menampung hasil tokening input text |
| 9 | Inisialisasi variabel untuk menampung data abstrak |
| 10-11 | Inisialisasi variabel |
| 12-26 | Perulangan untuk menampung hasil praproses text untuk data abstrak pada database |
| 27 | Memanggil fungsi <code>tags_to_point</code> pada kelas <code>Similarity</code> untuk menyimpan kata-kata unik yang terdapat pada data abstrak dan memberinya nilai 0. |
| 28 | Inisialisasi variabel untuk menampung data hasil token input text dan memberinya nilai 1. |
| 29-41 | Perulangan untuk menghitung kesamaan text antara input text dengan data bstrak pada database dengan memanggil fungsi <code>cosine</code> pada kelas <code>similarity</code> . |
| 42 | Menghitung total persentase kemiripan text |
| 43-44 | load view |

5.2.3.2 Implementasi Kode Program Kelas admin

Nama Fungsi : `rekomendasiJuri()`

Source Code :

Kode Program 5.2 Kode Program fungsi rekomendasiJuri pada kelas admin

| | |
|----|--|
| 1 | <code>public function rekomendasiJuri(){</code> |
| 2 | <code> \$karyawan=\$this->model_db->selectAll('karyawan');</code> |
| 3 | <code> \$pelatihan=\$this->model_db->selectAll('pelatihan');</code> |
| 4 | <code> \$i=0;</code> |
| 5 | <code> foreach(\$karyawan as \$value){</code> |
| 6 | <code> foreach(\$pelatihan as \$value1){</code> |
| 7 | <code> if(\$value->npk == \$value1->npk){</code> |
| 8 | <code> if(\$value1->kriteria == 1){</code> |
| 9 | <code> \$c1=1; \$c2=1; \$c3=1;</code> |
| 10 | <code> foreach(\$pelatihan as \$value2){</code> |
| 11 | <code> if(\$value1->npk == \$value2->npk){</code> |
| 12 | <code> if(\$value2->kriteria == 2){</code> |
| 13 | <code> \$c1 = \$c1+1;</code> |
| 14 | <code> }</code> |
| 15 | <code> else if(\$value2->kriteria == 3){</code> |
| 16 | <code> \$c2 = \$c2+1;</code> |
| 17 | <code> }</code> |
| 18 | <code> else if(\$value2->kriteria == 4){</code> |
| 19 | <code> \$c3 = \$c3+1;</code> |
| 20 | <code> }</code> |
| 21 | <code> }</code> |
| 22 | <code> }</code> |
| 23 | <code> \$bobotKriteria = array(</code> |
| 24 | <code> "C1" => \$c1,</code> |
| 25 | <code> "C2" => \$c2,</code> |
| 26 | <code> "C3" => \$c3</code> |
| 27 | <code>);</code> |
| 28 | <code> \$calonJuri[\$i] = array(</code> |
| 29 | <code> "npk" => \$value->npk,</code> |
| 30 | <code> "nama" => \$value->nama,</code> |
| 31 | <code> "bobotKriteria" => \$bobotKriteria</code> |
| 32 | <code>);</code> |
| 33 | <code> \$i++;</code> |

| | |
|----|---|
| 34 | } |
| 35 | } |
| 36 | } |
| 37 | } |
| 38 | \$vi = weightedProduct::countVi(\$calonJuri); |
| 39 | for(\$i = 0; \$i < sizeof(\$calonJuri); \$i++){ |
| 40 | \$data['bobotKriteria'][\$i]= |
| 41 | \$calonJuri[\$i]['bobotKriteria']; |
| 42 | \$data['nama'][\$i] = \$calonJuri[\$i]['nama']; |
| 43 | \$data['npk'][\$i] = \$calonJuri[\$i]['npk']; |
| 44 | \$data['vi'][\$i] = \$vi[\$i]; |
| 45 | } |
| 46 | \$data['admin']=\$this->model_db->selectAll('admin'); |
| 47 | \$this->load->view('hasilRekomendasi',\$data); |
| 48 | } |

Penjelasan :

Tabel 5.18 penjelasan kode program fungsi rekomendasiJuri pada kelas admin

| | |
|-------|--|
| 1 | Deklarasi fungsi rekomendasiJuri |
| 2 | Inisialisasi variabel untuk menampung data karyawan |
| 3 | Inisialisasi variabel untuk menampung data pelatihan |
| 4 | Inisialisasi variabel |
| 5-37 | Perulangan untuk menampung data npk, nama, dan bobot kriteria dari setiap juri |
| 38 | Memanggil fungsi countVi pada kelas weightedProduct untuk menghitung nilai vi |
| 39-45 | Perulangan untuk menyimpan data calon juri yang telah diberi nilai vi |
| 46 | Inisialisasi variabel untuk menampung data admin |
| 47 | Load view |

5.2.3.3 Implementasi Kode Program Kelas Juri

Nama Fungsi : menilai()

Source Code :

Kode Program 5.3 Kode Program fungsi menilai pada kelas juri

| | |
|----|---|
| 1 | public function menilai(\$nomorGugus){ |
| 2 | \$nilai = \$this->input->post('nilai'); |
| 3 | \$bobotSS_SSG = [8,5,8,12,4,4,8,8,12,8,8,5,8,2]; |
| 4 | \$bobotFAME = [10,15,10,12,8,12,8,10,5,8,2]; |
| 5 | \$bobotPKM_GKM = |
| 6 | [7,4,1,6,6,6,6,1,3,6,6,4,5,8,3,3,3,4,3,5,4,4,2]; |
| 7 | \$jumlah = 0; |
| 8 | \$hasil = 0; |
| 9 | \$hasilAkhir = 0; |
| 10 | \$data['nomor_gugus'] = \$nomorGugus; |
| 11 | \$pendaftaran=\$this->model_db-> |
| 12 | >get_data('pendaftaran',\$data)->row(); |
| 13 | \$data['juri'] = \$this->model_db->selectAll('juri'); |
| 14 | if(\$pendaftaran->jenis_gugus == "SSG" |
| 15 | \$pendaftaran->jenis_gugus == "SS"){ |
| 16 | \$i = 0; |
| 17 | foreach(\$nilai as \$value){ |
| 18 | \$hasil = \$hasil + |
| 19 | (\$value*\$bobotSS_SSG[\$i]); |
| 20 | \$i++; |

| | |
|----|--|
| 21 | } |
| 22 | } |
| 23 | else if(\$pendaftaran->jenis_gugus == "FAME"){ |
| 24 | \$i = 0; |
| 25 | foreach(\$nilai as \$value){ |
| 26 | \$hasil = \$hasil + |
| 27 | (\$value*\$bobotFAME[\$i]); |
| 28 | \$i++; |
| 29 | } |
| 30 | } |
| 31 | else if(\$pendaftaran->jenis_gugus == "PKM" |
| 32 | \$pendaftaran->jenis_gugus == "GKM"){ |
| 33 | \$i = 0; |
| 34 | foreach(\$nilai as \$value){ |
| 35 | \$hasil = \$hasil + |
| 36 | (\$value*\$bobotPKM_GKM[\$i]); |
| 37 | \$i++; |
| 38 | } |
| 39 | } |
| 40 | \$data['npk_juri'] = \$_SESSION['npk']; |
| 41 | \$data['nomor_gugus'] = \$nomorGugus; |
| 42 | \$data['nilai'] = \$hasil; |
| 43 | \$this->model_db->add('penilaian', \$data); |
| 44 | \$penilaian = \$this->model_db-> |
| 45 | selectAll('penilaian'); |
| 46 | foreach(\$penilaian as \$value){ |
| 47 | if(\$value->nomor_gugus == \$nomorGugus){ |
| 48 | \$jumlah = \$jumlah+1; |
| 49 | \$hasilAkhir = \$hasilAkhir + \$value-> |
| 50 | nilai; |
| 51 | } |
| 52 | } |
| 53 | \$hasilAkhir = \$hasilAkhir/\$jumlah; |
| 54 | \$data['hasil'] = \$hasil; |
| 55 | \$data['hasilAkhir'] = \$hasilAkhir; |
| 56 | \$this->load->view('hasilNilai',\$data); |
| 57 | } |

Penjelasan :

Tabel 5.19 Penjelasan kode program fungsi menilai pada kelas juri

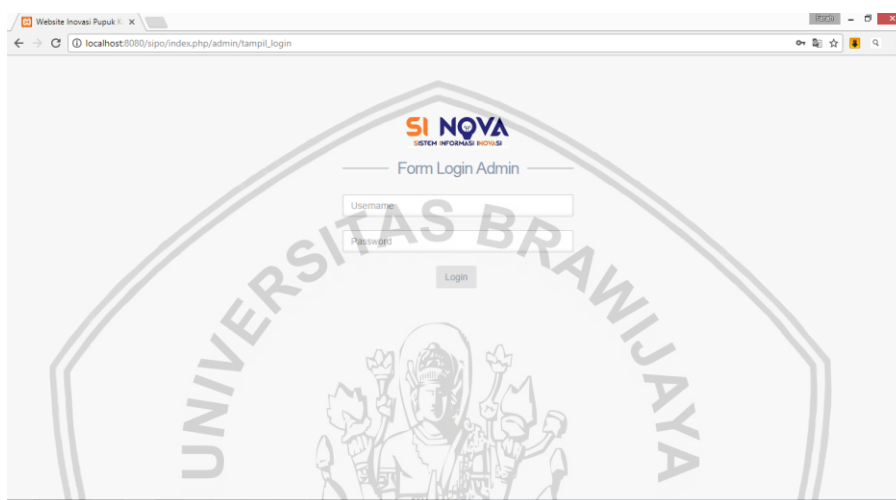
| | |
|-------|---|
| 1 | Deklarasi fungsi menilai |
| 2 | Inisialisasi variabel untuk menampung input nilai |
| 3-6 | Inisialisasi variabel untuk menampung bobot setiap jenis gugus |
| 7-13 | Inisialisasi variabel |
| 14-39 | Perhitungan nilai dari setiap jenis gugus dengan mengalikan setiap kategori dari input nilai dengan bobot dari setiap jenis gugus |
| 40-45 | Inisialisasi variabel |
| 46-52 | Perulangan untuk menghitung hasil akhir nilai dari setiap gugus dengan menjumlahkan semua nilai yang diberikan juri-juri untuk setiap gugus dan menghitung jumlah juri yang telah memberikan nilai pada suatu gugus |
| 53 | Menghitung hasil akhr dengan menghitung jumlah nilai yang didapat setiap gugus dan membagi dengan jumlah juri yang telah menilai. |
| 54-55 | Inisialisasi variabel |
| 56 | Load view |

5.2.4 Implementasi Antarmuka

Pada implementasi antarmuka sistem terdapat beberapa sampel impelementasi antarmuka sistem seperti halaman utama, halaman login, halaman home, halaman cek plagiasi, halaman rekomendasi sistem, dll. Implementasi antarmuka sistem pada sub bab ini dibangun berdasarkan rancangan antarmuka sistem yang sudah dijelaskan pada sub bab perancangan antarmuka sistem.

5.2.4.1 Implementasi Antarmuka Halaman Login

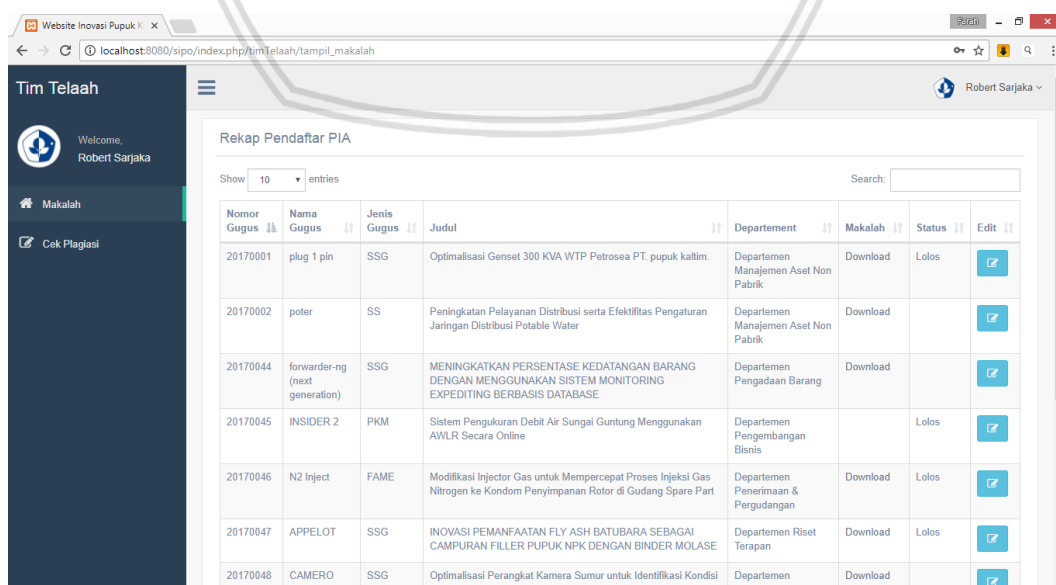
Implementasi antarmuka halaman login pada sistem penjurian online ditunjukkan pada Gambar 5.20.



Gambar 5.19 Implementasi Antarmuka Halaman Login

5.2.4.2 Implementasi Antarmuka Halaman Home Tim Telaah

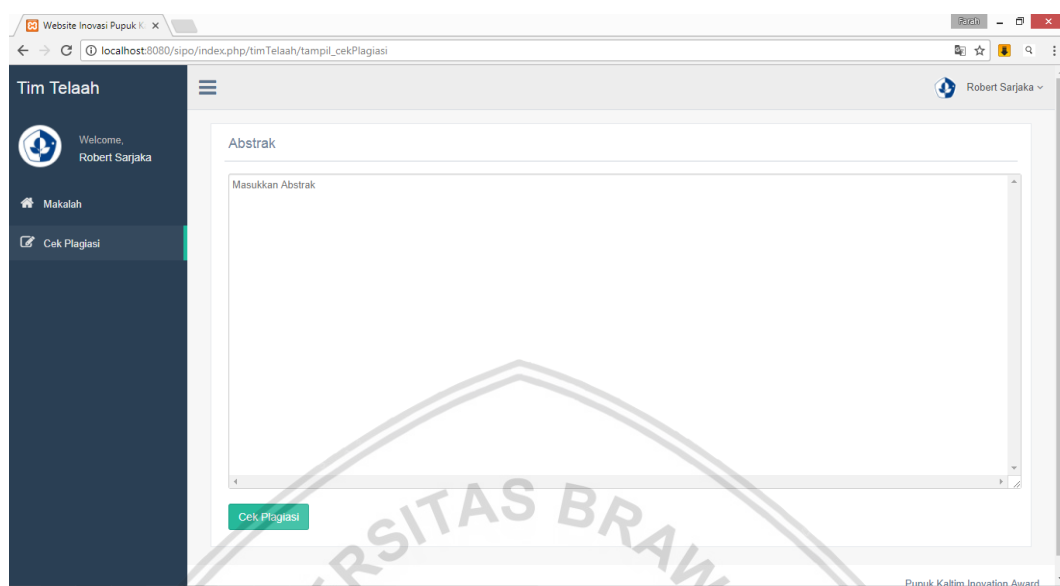
Implementasi antarmuka halaman home tim telaah pada sistem penjurian online ditunjukkan pada Gambar 5.21.



Gambar 5.20 Implementasi Halaman Home Tim Telaah.

5.2.4.3 Implementasi Antarmuka Halaman Cek Plagiasi

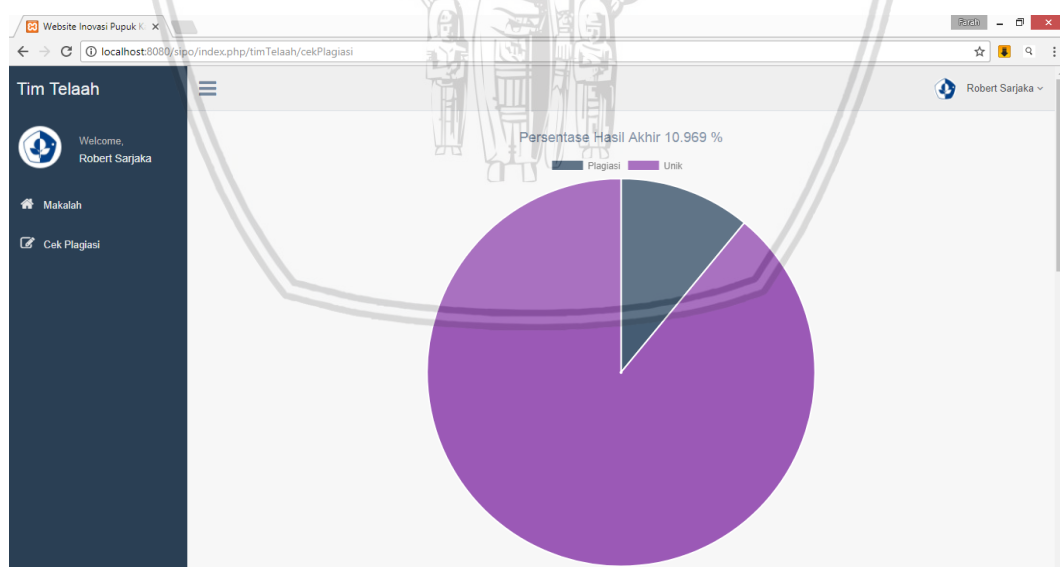
Implementasi antarmuka halaman cek plagiasi pada sistem penjurian online ditunjukkan pada Gambar 5.22.



Gambar 5.21 Implementasi Antarmuka Halaman Cek Plagiasi

5.2.4.4 Implementasi Antarmuka Halaman Hasil Cek Plagiasi

Implementasi antarmuka halaman hasil cek plagiasi pada sistem penjurian online ditunjukkan pada Gambar 5.23.



| Hasil Cek Plagiasi | | |
|--------------------|---|------------------|
| No | Judul | Persentase |
| 1 | PROSES ADMINISTRASI DENGAN SISTEM REIMBURSE DALAM PEMBERIAN BANTUAN KEPADA PIHAK KETIGA | 9.6885155812286% |
| 2 | Publikasi Perusahaan melalui Media Online | 15.238807260007% |
| 3 | Penggunaan Velcro untuk Penataan Kabel | 11.305730797794% |
| 4 | PEMANFAATAN APLIKASI WHATS APP SEBAGAI ALAT KOORDINASI SEKRETARIS DIREKSI DALAM PENYUSUNAN AGENDA MANAJEMEN | 15.403080924308% |
| 5 | PEMANFAATAN APLIKASI WHATS APP SEBAGAI ALAT PENGIRIMAN MEDIA DAN DOKUMEN | 10.870706918483% |
| 6 | Pengelolaan Surat dan Dokumen Aktif Perusahaan dengan Pengembangan Sistem Informasi Kearsipan Dinamis PT Pupuk Kalimantan Timur (SIKD PKT) | 11.708833169601% |
| 7 | Klasifikasi Temuan Dalam Mempercepat Tindak Lanjut Hasil Audit | 13.805369799253% |
| 8 | Meningkatkan Budaya Sadar Risiko dengan Menciptakan dan Mengimplementasikan Sistem Informasi Laporan Manajemen Risiko Individu (Risk) | 13.111853491496% |
| 9 | Pembuatan Aplikasi Otomatisasi Pairing Penilai-Dinilai pada Proses Penilaian 360 | 5.7977103565245% |
| 10 | DIGITALISASI & INTEGRASI PROSES BISNIS SDM DENGAN ERP PUPUK KALTIM UNTUK MENINGKATKAN EFISIENSI WAKTU PENYUSUNAN JOB DESCRIPTION & KOMPETENSI | 14.329954416637% |
| 11 | MENINGKATKAN PRODUKTIVITAS KARYAWAN LAPANGAN, DENGAN APLIKASI SISTEM ABSENSI BERBASIS MOBILE ANDROID | 7.1210911486986% |
| 12 | PENGUNAAN TABEL DALAM PEMBUATAN GRAFIK KATEGORI 7 KRITERIA PENILAIAN KINERJA UNGGUL | 8.8929729179989% |
| 13 | Mempercepat proses penyiapan formulir sertifikasi sesuai ceklis di TUK Pupuk Kaltim sebesar 100 % | 10.922639774891% |
| 14 | Efisiensi Biaya Form Pendaftaran Pupuk Kaltim Innovation Award XXIX Tahun 2017 | 8.4366148773211% |
| 15 | Peningkatan Kualitas Pengamanan Area PT Pupuk Kalimantan Timur, Tahun 2017 | 11.425405340195% |

Gambar 5.22 Implementasi Antarmuka Halaman Hasil Cek Plagiasi

5.2.4.5 Implementasi Antarmuka Halaman Home Admin

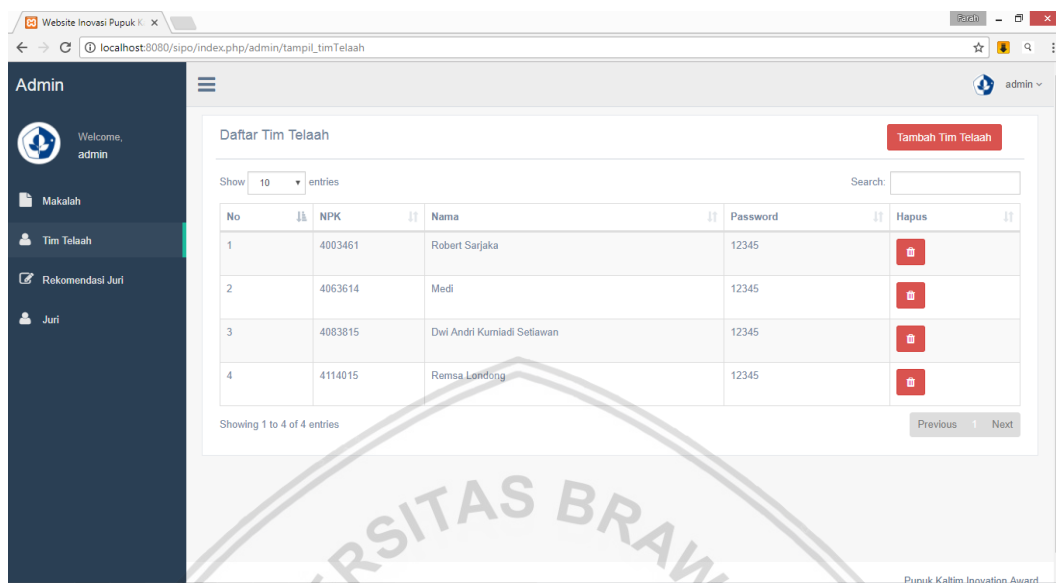
Implementasi antarmuka halaman home admin pada sistem penjurian online ditunjukkan pada Gambar 5.24.

| Rekap Makalah PIA Lolos | | | | | | | | |
|-------------------------|------------|-------------|--|--------------------------------------|----------|--------|--------|--------|
| Nomor Gugus | Nama Gugus | Jenis Gugus | Judul | Departemen | Makalah | Stream | Status | Action |
| 20170001 | plug 1 pin | SSG | Optimalisasi Genset 300 KVA WTP Petrosas PT. Pupuk Kaltim | Departemen Manajemen Aset Non Pabrik | Download | 1 | Lolos | |
| 20170045 | INSIDER 2 | PKM | Sistem Pengukuran Debit Air Sungai Guntung Menggunakan AWLR Secara Online | Departemen Pengembangan Bisnis | | 1 | Lolos | |
| 20170046 | N2 Inject | FAME | Modifikasi Injector Gas untuk Mempercepat Proses Injeksi Gas Nitrogen ke Kondom Penyimpanan Rotor di Gudang Spare Part | Departemen Penerimaan & Pergudangan | Download | 1 | Lolos | |
| 20170047 | APPELOT | SSG | INOVASI PEMANFAATAN FLY ASH BATUBARA SEBAGAI CAMPURAN FILLER PUPUK NPK DENGAN BINDER MOLASE | Departemen Riset Terapan | Download | 1 | Lolos | |

Gambar 5.23 Implementasi Antarmuka Halaman Home Admin

5.2.4.6 Implementasi Antarmuka Halaman Daftar Tim Telaah

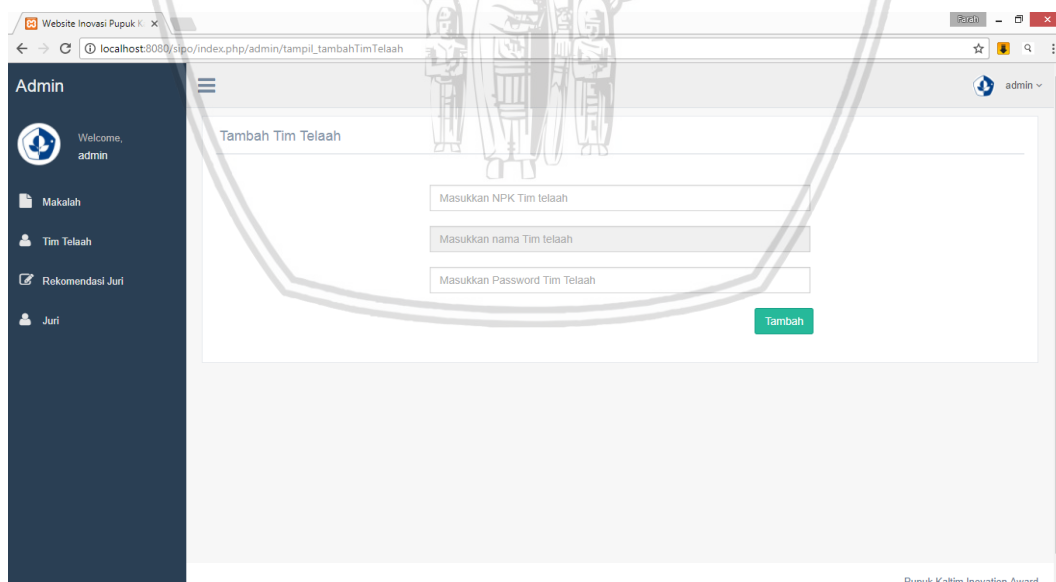
Implementasi antarmuka halaman daftar tim telaah pada sistem penjurian online ditunjukkan pada Gambar 5.25.



Gambar 5.24 Implementasi Antarmuka Halaman Daftar Tim Telaah

5.2.4.7 Implementasi Antarmuka Halaman Tambah Tim Telaah

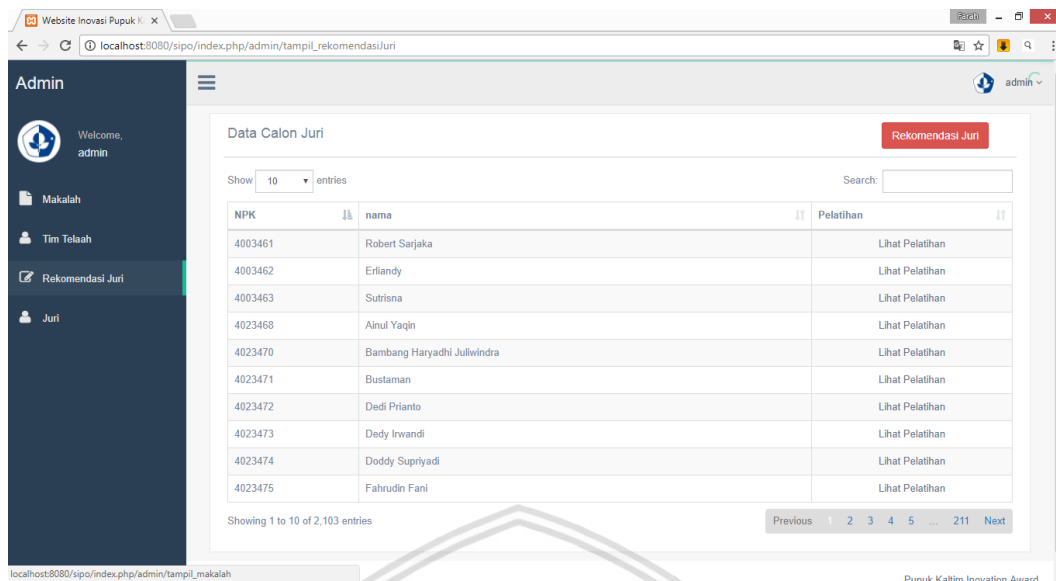
Implementasi antarmuka halaman tambah tim telaah pada sistem penjurian online ditunjukkan pada Gambar 5.26.



Gambar 5.25 Implementasi Antarmuka Halaman Tambah Tim Telaah

5.2.4.8 Implementasi Antarmuka Halaman Data Calon Juri

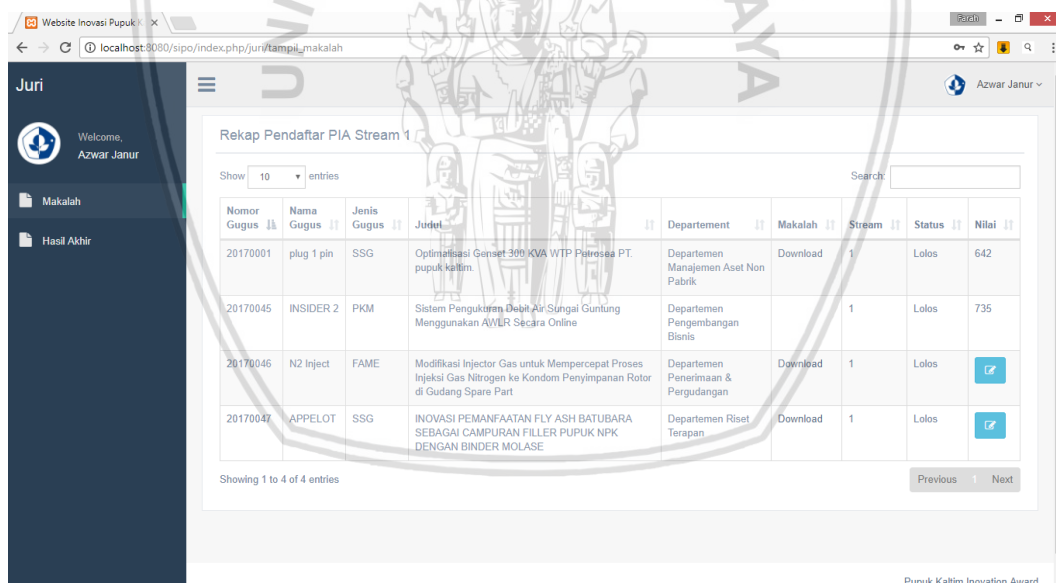
Implementasi antarmuka halaman data calon juri pada sistem penjurian online ditunjukkan pada Gambar 5.27.



Gambar 5.26 Implementasi Antarmuka Halaman Data Calon Juri

5.2.4.9 Implementasi Antarmuka Halaman Home Juri

Implementasi antarmuka halaman home juri pada sistem penjurian online ditunjukkan pada Gambar 5.28.



Gambar 5.27 Implementai Antarmuka Halaman Home Juri

5.2.4.10 Implementasi Antarmuka Halaman Form Penilaian

Implementasi antarmuka halaman form penilaian pada sistem penjurian online ditunjukkan pada Gambar 5.29.

Form Penilaian FAME

Plan

Permasalahan

Pemilihan Masalah 1 2 3 4 5 6 7 8 9 10

Analisis Akar Penyebab

Pencarian Ide 1 2 3 4 5 6 7 8 9 10

Rencana Penanggulangan

Penetapan Penyebab Domain Masalah 1 2 3 4 5 6 7 8 9 10

Perencanaan perbaikan 1 2 3 4 5 6 7 8 9 10

Do

Uji Coba Pelaksanaan Perbaikan

Tahap Uji Coba 1 2 3 4 5 6 7 8 9 10

Check

Penelitian Hasil Perbaikan

Gambar 5.28 Implementasi Antarmuka Halaman *Form* Penilaian

5.2.4.11 Implementasi Antarmuka Halaman Hasil Akhir Penilaian

Implementasi antarmuka halaman hasil akhir penilaian pada sistem penjurian online ditunjukkan pada Gambar 5.30.

Report Hasil Akhir Penilaian

Stream 4 Stream 3 Stream 2 Stream 1

Stream 1

Show 10 entries Search:

| Nomor Gugus | Nama Gugus | Jenis Gugus | Judul | Departemen | Makalah | Stream | Status | Nilai |
|-------------|------------|-------------|--|--------------------------------------|----------|--------|--------|-------|
| 20170001 | plug 1 pin | SSG | Optimalisasi Genset 300 KVA WTP Petrosea PT. pupuk kaltim. | Departemen Manajemen Aset Non Pabrik | Download | 1 | Lolos | 0 |
| 20170045 | INSIDER 2 | PKM | Sistem Pengukuran Debit Air Sungai Guntung Menggunakan AWLR Secara Online | Departemen Pengembangan Bisnis | | 1 | Lolos | 735 |
| 20170046 | N2 Inject | FAME | Modifikasi Injector Gas untuk Mempercepat Proses Injeksi Gas Nitrogen ke Kondom Penyimpanan Rotor di Gudang Spare Part | Departemen Penerimaan & Pergudangan | Download | 1 | Lolos | 0 |
| 20170047 | APPELOT | SSG | INOVASI PEMANFAATAN FLY ASH BATUBARA SEBAGAI CAMPURAN FILLER PUPUK NPK DENGAN BINDER MOLASE | Departemen Riset Terapan | Download | 1 | Lolos | 0 |

Showing 1 to 4 of 4 entries Previous Next

Gambar 5.29 Implementasi Antarmuka Halaman Hasil Akhir Penilaian

BAB 6 PENGUJIAN

Tahap pengujian dilakukan setelah tahap implementasi telah selesai. Pengujian bertujuan untuk memeriksa apakah implementasi sudah sesuai dengan analisis kebutuhan dan perancangan sistem atau tidak. Tahap pengujian yang akan dilakukan adalah pengujian unit, pengujian integrasi, dan pengujian validasi.

6.1 Pengujian Unit

Pengujian unit adalah salah satu metode pengujian yang dilakukan dengan cara fokus pada pengujian setiap unit dari sistem yang telah diimplementasikan seperti: komponen sistem, kelas, atau objek. Pengujian unit merupakan bagian dari metode pengujian *whitebox* yang mana menggunakan teknik pengujian *basis path*. Metode ini digunakan untuk mengukur kompleksitas logika dari perancangan prosedural yang digunakan dalam pengembangan sistem. Langkah-langkah yang dilakukan dalam melakukan pengujian unit ini adalah:

1. Pembuatan *flow graph* dari perancangan prosedural kode program (*pseudocode*).
2. Menentukan *Cyclomatic complexity* dari *flow graph*.
3. Menentukan *Independent path* dari *flow graph*.

6.1.2 Pengujian Unit kelas *timTelaah* operasi *cekPlagiasi()*

Pada pengujian unit kelas *timTelaah* operasi *cekPlagiasi()* terdapat 2 langkah yang dilakukan yaitu:

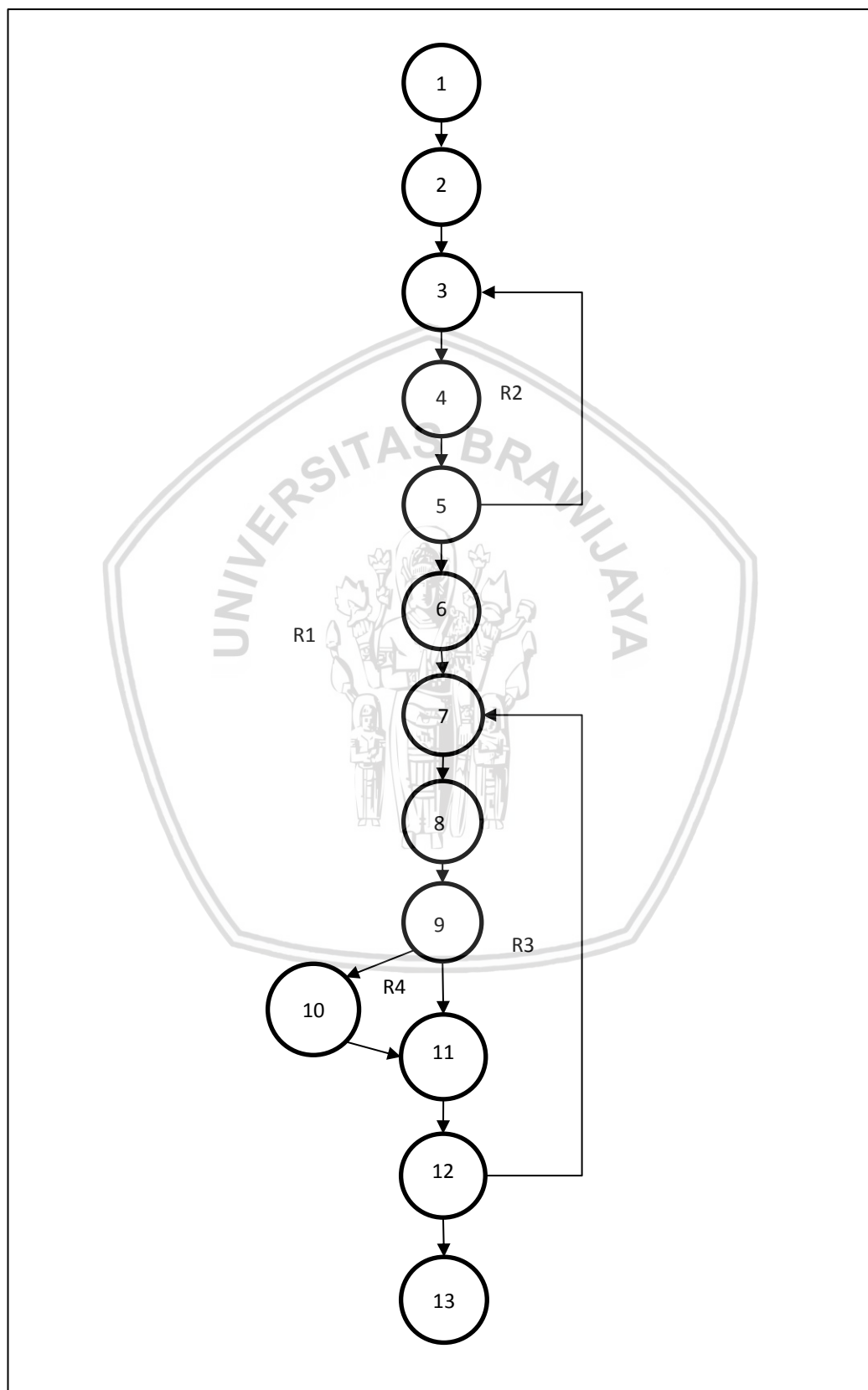
1. *Pseudocode*

```

Start (1)
Text = input (abstrak)
stemms = hasil stemming text input
tokens = hasil tokenisasi text input
total = banyak data abstrak
jumlah = 0
pembagi = 0
for i to total Do (3)
    text1 = get abstrak ke i
    stemms1 = hasil stemming text1
    tokens1 = hasil tokenisasi text1
    articles[i] = tokens1
endfor (5)
tags = memberi nilai pada setiap data tags
compare = memberi nilai setiap value array tokens bernilai 1 dan menampung value tags (6)
for i to total Do (7)
    ak = memberi nilai setiap value array articles ke i dengan nilai 1 dan menampung tags
    hasil[i] = menyimpan data hasil proses perhitungan cosine similarity data inputan dengan setiap data abstrak pada basis data (8)
    jumlah += hasil[i]
    if hasil[i] != 0 (9)
        pembagi++ (10)
    endif (11)
endfor (12)
end (13)
    
```

2. Basis Path Testing

a. Flow Graph



b. *Cyclomatic Complexity*

- $V(G) = 4$, ada 4 region R1, R2, R3, R4
- $V(G) = 15 \text{ edges} - 13 \text{ nodes} + 2 = 4$
- $V(G) = 3 \text{ predicate nodes} + 1 = 4$

c. *Independent Path*

- Jalur 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13
- Jalur 2 = 1 – 2 – 3 – 4 – 5 – 3...
- Jalur 3 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 7...
- Jalur 4 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 11 – 12 – 13

Test case dan hasil akan dijelaskan pada tabel 6.1 dibawah ini.

Tabel 6.1 Hasil Pengujian Unit Kelas timTelaah Operasi cekPlagiasi()

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|---|--|--|--------|
| 1. | 1 | Valid input, dimana $i < \text{total data abstrak}$ dan $\text{hasil}[i] = 1$ | Operasi akan mengembalikan nilai TRUE | Operasi berhasil mengembalikan nilai TRUE | valid |
| 2. | 2 | Valid input, dimana $i < \text{total data abstrak}$ | Operasi akan meneruskan perulangan | Operasi berhasil meneruskan perulangan | valid |
| 3. | 3 | Valid input, dimana $i < \text{total data abstrak}$ | Operasi akan meneruskan perulangan | Operasi berhasil meneruskan perulangan | valid |
| 4. | 4 | Invalid input, dimana $\text{hasil}[i] = 0$ | Operasi akan mengembalikan nilai FALSE | Operasi berhasil mengembalikan nilai FALSE | valid |

5.1.2 Pengujian Unit Kelas admin Operasi rekomendasiJuri()

Pada pengujian unit kelas admin operasi `rekomendasiJuri()` terdapat 2 langkah yang dilakukan yaitu:

1. *Pseudocode*

```

Start (1)
karyawan = data karyawan
pelatihan = data pelatihan } (2)
For i to jumlah karyawan Do (3)
    For j to jumlah pelatihan Do (4)
        If npk_karyawan[i] == npk_pelatihan[j] (5)
            If kriteria_pelatihan[j] == 1 (6)
    
```

```

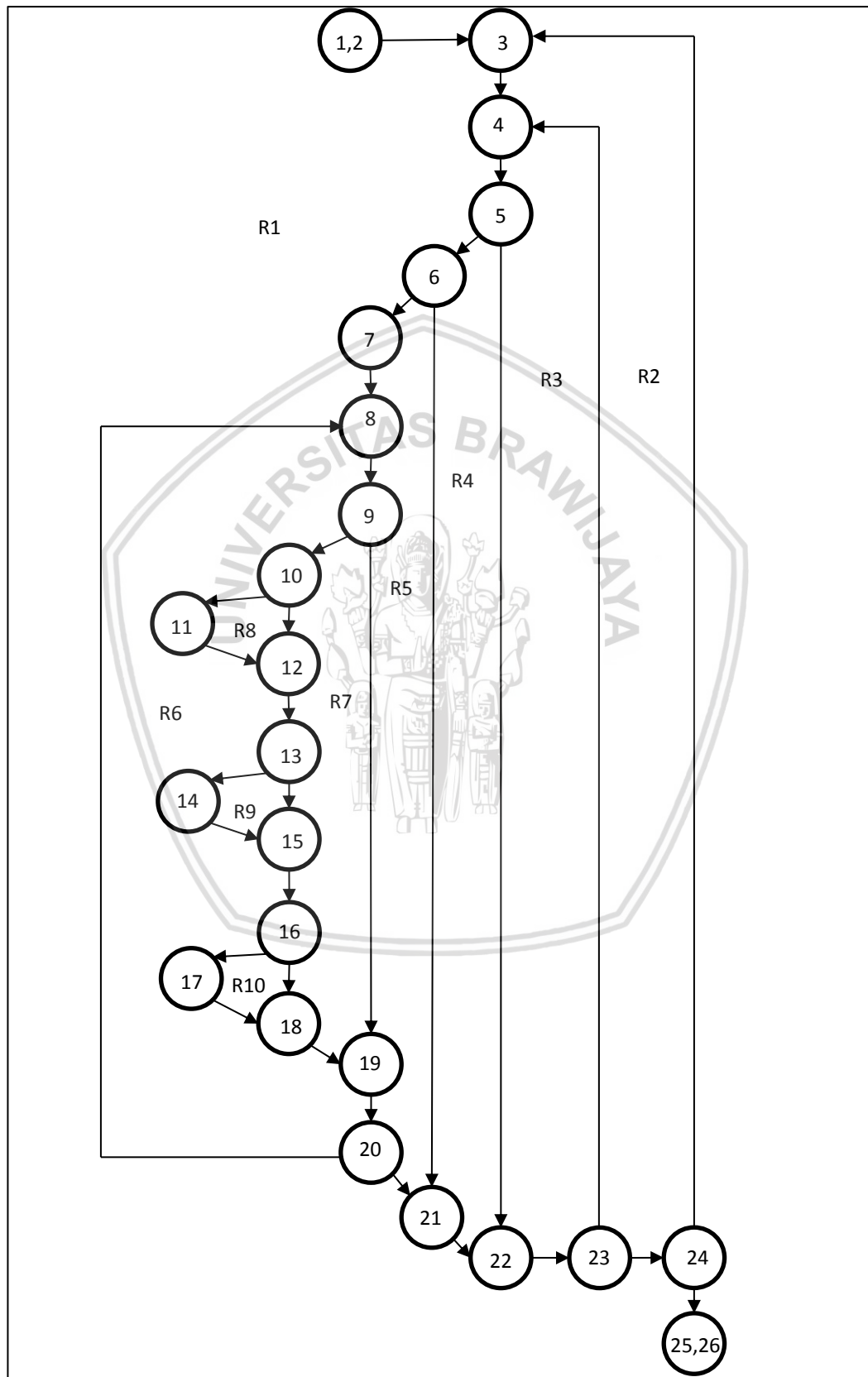
C1 = 1
C2 = 1
C3 = 1
Foreach k to jumlah pelatihan Do (8)
  If npk_pelatihan[j] == npk_pelatihan[k] (9)
    If kriteria_pelatihan[k] == 2 (10)
      C1 = C1 + 1 (11)
    endif (12)
    else if kriteria_pelatihan[k] == 3 (13)
      C2 = C2 + 1 (14)
    endif (15)
    else if kriteria_pelatihan[k] == 4 (16)
      C3 = C3 + 1 (17)
    endif (18)
  endif (19)
endfor (20)
endif (21)
endif (22)
endfor (23)
endfor (24)
BobotKriteria = [C1, C2, C3]
calonJuri = data calon juri
vi = hasil perhitungan dengan metode weighted product
end (26)
} (25)

```



2. Basis Path Testing

a. Flow Graph



b. *Cyclometric Complexity*

- $V(G) = 10$, ada 10 region R1, R2, R3, R4, R5, R6, R7, R8, R9, R10
- $V(G) = 32 \text{ edges} - 24 \text{ nodes} + 2 = 10$
- $V(G) = 9 \text{ predicate nodes} + 1 = 10$

c. *Independent Path*

- Jalur 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26
- Jalur 2 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 3...
- Jalur 3 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 4...
- Jalur 4 = 1 – 2 – 3 – 4 – 5 – 22 – 23 – 24 – 25 – 26
- Jalur 5 = 1 – 2 – 3 – 4 – 5 – 6 – 21 – 22 – 23 – 24 – 25 – 26
- Jalur 6 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 8...
- Jalur 7 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26
- Jalur 8 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26
- Jalur 9 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26
- Jalur 10 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 18 – 19 – 20 – 21 – 22 – 23 – 24 – 25 – 26

Test case dan hasil akan dijelaskan pada tabel 6.2 dibawah ini.

Tabel 6.2 Hasil Pengujian Unit Kelas Admin Operasi rekomendasiJuri()

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|---|---|---|--------|
| 1. | 1 | Valid input, dimana $i < \text{jumlah karyawan dan kriteria_pelatihan}[j] = 1$ | Operasi akan mengembalikan nilai TRUE dan C1, C2, C3 bernilai 1 | Operasi berhasil mengembalikan nilai TRUE dan C1, C2, C3 bernilai 1 | valid |
| 2. | 2 | Valid input, dimana $i < \text{jumlah karyawan}$ | Operasi akan meneruskan perulangan | Operasi berhasil meneruskan perulangan | valid |

| | | | | | |
|-----|----|--|---|---|-------|
| 3. | 3 | Valid input, dimana $i < \text{jumlah pelatihan}$ | Operasi akan meneruskan perulangan | Operasi berhasil meneruskan perulangan | valid |
| 4. | 4 | Invalid input, dimana $\text{npk_karyawan}[i] \neq \text{npk_pelathan}[j]$ | Operasi akan mengembalikan nilai FALSE | Operasi berhasil mengembalikan nilai FALSE | valid |
| 5. | 5 | Invalid input, dimana $\text{kriteria_pelatihan}[j] = 2$ | Operasi akan mengembalikan nilai FALSE | Operasi berhasil mengembalikan nilai FALSE | valid |
| 6. | 6 | Valid input, dimana $k < \text{jumlah pelatihan}$ | Operasi akan meneruskan perulangan | Operasi berhasil meneruskan perulangan | valid |
| 7. | 7 | Invalid input, dimana $\text{npk_karyawan}[j] \neq \text{npk_pelathan}[k]$ | Operasi akan mengembalikan nilai FALSE | Operasi berhasil mengembalikan nilai FALSE | valid |
| 8. | 8 | Invalid input, dimana $\text{kriteria_pelatihan}[k] = 4$ | Operasi akan mengembalikan nilai FALSE dan menambahkan 1 nilai dari variabel C3 | Operasi berhasil mengembalikan nilai FALSE dan menambahkan 1 nilai dari variabel C3 | valid |
| 9. | 9 | Invalid input, dimana $\text{kriteria_pelatihan}[j] = 2$ | Operasi akan mengembalikan nilai FALSE dan menambahkan 1 nilai dari variabel C1 | Operasi berhasil mengembalikan nilai FALSE dan menambahkan 1 nilai dari variabel C1 | valid |
| 10. | 10 | Invalid input, dimana $\text{kriteria_pelatihan}[j] = 3$ | Operasi akan mengembalikan nilai FALSE dan menambahkan 1 nilai dari variabel C2 | Operasi berhasil mengembalikan nilai FALSE dan menambahkan 1 nilai dari variabel C2 | valid |

6.1.3 Pengujian Unit Kelas juri Operasi menilai()

Pada pengujian unit kelas juri operasi menilai() terdapat 2 langkah yang dilakukan yaitu:

1. Pseudocode

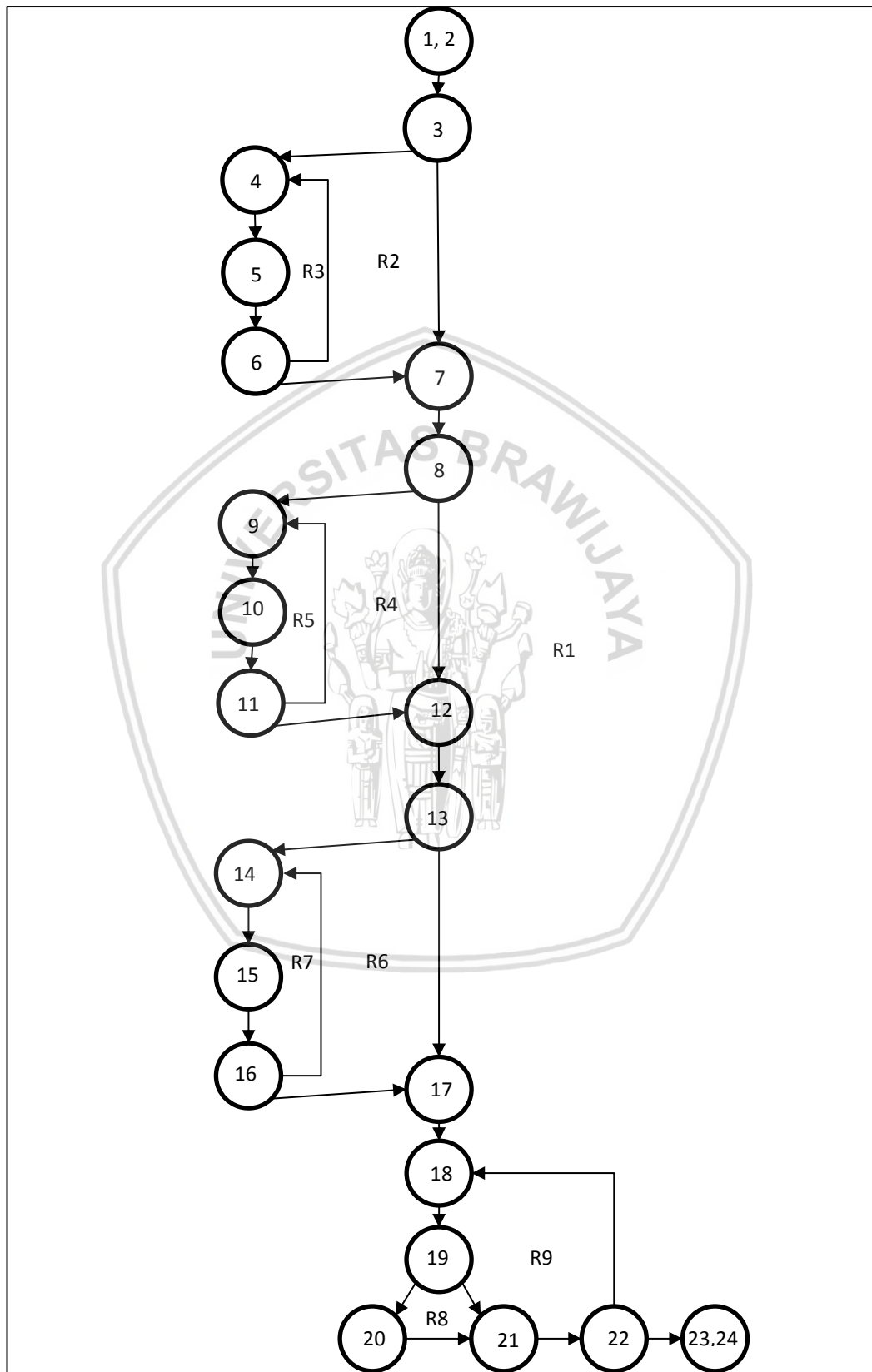
```

Start (1)
nomorGugus = parameter
Nilai = input(nilai)
bobotSS_SSG = bobot nilai makalah jenis SS dan SSG
bobotFAME = bobot nilai makalah jenis FAME
boboPKM_GKM = bobot nilai makalah jenis PKM dan GKM
jumlah = 0
hasil = 0
hasilAkhir = 0
Pendaftaran = data pendaftaran
Penilaian = data penilaian
If (jenis gugus pendaftaran = SSG atau SS) (3)
    for i in jumlah nilai Do (4)
        Hasil = hasil + (nilai[i]*bobotSS_SSG[i]) (5)
    endfor (6)
endif (7)
else if (jenis gugus pendaftaran = FAME) (8)
    for i in jumlah nilai Do (9)
        Hasil = hasil + (nilai[i]*bobotFAME[i]) (10)
    endfor (11)
endif (12)
else if (jenis gugus pendaftaran = PKM atau GKM) (13)
    for i in jumlah nilai Do (14)
        Hasil = hasil + (nilai[i]*bobotPKM_GKM[i]) (15)
    endfor (16)
endif (17)
for i in jumlah penilaian Do (18)
    if nomor gugus pada data penilaian = nomorGugus (19)
        Jumlah = jumlah+1
        hasilAkhir = hasilAkhir + nilai pada data penilaian } (20)
    endif (21)
endfor (22)
hasilAkhir = hasilAkhir/jumlah (23)
end (24)

```


2. Basis Path Testing

a. Flow Graph



b. *Cyclomatic Complexity*

- $V(G) = 9$, ada 9 region R1, R2, R3, R4, R5, R6, R7, R8
- $V(G) = 29 \text{ edges} - 22 \text{ nodes} + 2 = 9$
- $V(G) = 8 \text{ predicate nodes} + 1 = 9$

c. *Independet Path*

- Jalur 1 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24
- Jalur 2 = 1 – 2 – 3 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24
- Jalur 3 = 1 – 2 – 3 – 4 – 5 – 6 – 4...
- Jalur 4 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24
- Jalur 5 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 9...
- Jalur 6 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 17 – 18 – 19 – 20 – 21 – 22 – 23 – 24
- Jalur 7 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 14...
- Jalur 8 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 21 – 22 – 23 – 24
- Jalur 9 = 1 – 2 – 3 – 4 – 5 – 6 – 7 – 8 – 9 – 10 – 11 – 12 – 13 – 14 – 15 – 16 – 17 – 18 – 19 – 20 – 21 – 22 – 18...

Test case dan hasil akan dijelaskan pada tabel 6.3 dibawah ini.

Tabel 6.3 Hasil Pengujian Unit Kelas juri Operasi menilai()

| No | No. Jalur | Prosedur Uji | Expected Result | Result | Status |
|----|-----------|---|---|---|--------|
| 1. | 1 | Valid input, dimana nomor gugus pada data penilaian = nomor gugus pada parameter fungsi | Operasi akan mengembalikan nilai TRUE, nilai dari variabel jumlah bertambah 1, dan variabel hasilAkhir ditambahkan dengan nilai hasil penilaian pada data ke[i] | Operasi berhasil mengembalikan nilai TRUE, nilai dari variabel jumlah bertambah 1, dan variabel hasilAkhir ditambahkan dengan nilai hasil penilaian pada data ke[i] | valid |

| | | | | | |
|----|---|---|---|---|-------|
| 2. | 2 | Invalid input, dimana jenis gugus = FAME | Operasi akan mengembalikan nilai FALSE dan memberi nilai variabel hasil dengan jumlah nilai yang diinputkan dengan bobotFAME | Operasi berhasil mengembalikan nilai FALSE dan memberi nilai variabel hasil dengan jumlah nilai yang diinputkan dengan bobotFAME | Valid |
| 3. | 3 | Valid input, dimana $i < \text{jumlah nilai}$ | Operasi akan meneruskan perulangan | Operasi berhasil meneruskan perulangan | valid |
| 4. | 4 | Invalid input, dimana jenis gugus = PKM | Operasi akan mengembalikan nilai FALSE dan memberi nilai variabel hasil dengan jumlah nilai yang diinputkan dengan bobotPKM_GKM | Operasi berhasil mengembalikan nilai FALSE dan memberi nilai variabel hasil dengan jumlah nilai yang diinputkan dengan bobotPKM_GKM | Valid |
| 5. | 5 | Valid input, dimana $i < \text{jumlah nilai}$ | Operasi akan meneruskan perulangan | Operasi berhasil meneruskan perulangan | valid |
| 6. | 6 | Invalid input, dimana jenis gugus = SSG | Operasi akan mengembalikan nilai FALSE dan memberi nilai variabel hasil dengan jumlah nilai yang diinputkan dengan bobotSS_SSG | Operasi berhasil mengembalikan nilai FALSE dan memberi nilai variabel hasil dengan jumlah nilai yang diinputkan dengan bobotSS_SSG | Valid |
| 7. | 7 | Valid input, dimana $i < \text{jumlah nilai}$ | Operasi akan meneruskan perulangan | Operasi berhasil meneruskan perulangan | valid |
| 8. | 8 | Valid input, dimana $i < \text{jumlah nilai}$ | Operasi akan meneruskan perulangan | Operasi berhasil meneruskan perulangan | valid |

| | | | | | |
|----|---|---|--|--|-------|
| | | jumlah penilaian | | | |
| 9. | 9 | Invalid input, nomor gugus pada data penilaian != nomor gugus pada parameter fungsi | Operasi akan mengembalikan nilai FALSE | Operasi berhasil mengembalikan nilai FALSE | valid |

6.2 Pengujian Integrasi

Pengujian integrasi merupakan metode pengujian dengan cara menguji unit-unit/komponen yang berinteraksi pada sistem. Pengujian integrasi ini memiliki tujuan untuk memeriksa fungsional, kinerja, dari struktur program yang dibangun. Apabila ditemukan hasil yang tidak sesuai dengan yang diharapkan maka integrasi antar kelas harus diperbaiki.

1. Pengujian Integrasi Method `cekPlagiasi()` pada kelas `timTelaah` dan Method `cosine()` pada kelas `similarity`

Method `cekPlagiasi()` pada kelas `timTelaah` merupakan method yang digunakan untuk mengolah data abstrak yang dimasukkan oleh pengguna dan data abstrak yang berada pada basis data sistem sehingga data tersebut dapat dibandingkan kemiripannya. Hasil pengolahan data tersebut kemudian akan dihitung kemiripannya menggunakan algoritme *cosine similarity* dengan memanggil method `cosine()` pada kelas `similarity`.

Tabel 6.4 Interaksi method `cekPlagiasi()` dengan method `cosine()`

| No. | Method dari kelas <code>timTelaah</code> | Method dari kelas <code>similarity</code> | Goal |
|-----|--|---|--|
| 1. | <code>cekPlagiasi()</code> | <code>cosine()</code> | Menghasilkan persentase kemiripan antara abstrak yang dimasukkan oleh pengguna dengan data abstrak yang berada pada basis data sistem. |

Prosedur uji yang dilakukan yaitu:

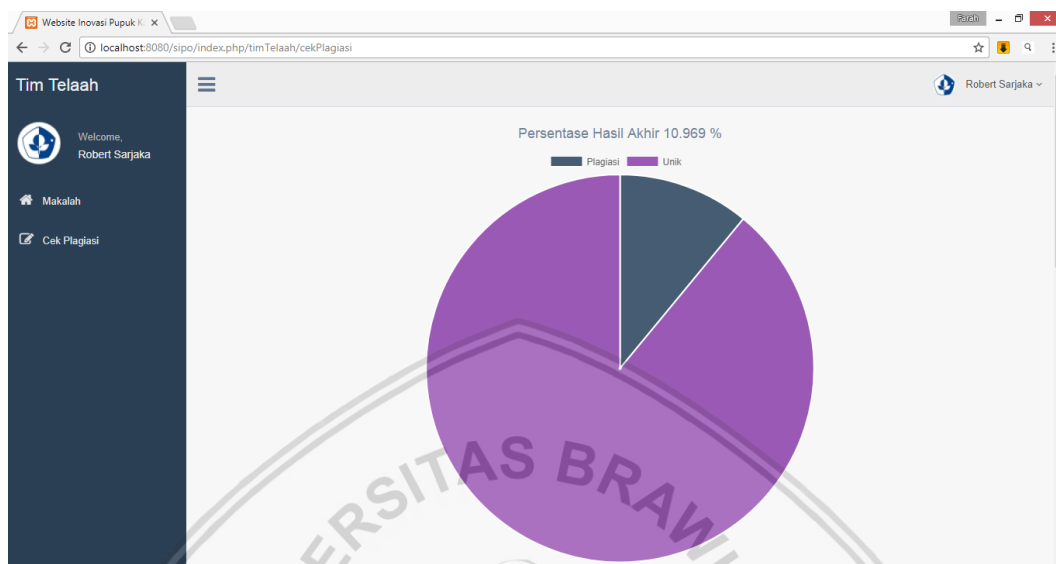
1. Menyiapkan kelas `timTelaah` untuk mengaktifkan method `cekPlagiasi()`.
2. Menyiapkan input-input yang diperlukan method `cekPlagiasi()`.
3. Mengeksekusi method `cekPlagiasi()` dan method `cosine()`.

Hasil Pengujian Integrasi Method `cekPlagiasi()` pada kelas `timTelaah` dan Method `cosine()` pada kelas `similarity` berdasarkan prosedur uji dapat dilihat pada Tabel 6.5.

Tabel 6.5 Hasil Pengujian Integrasi Method `cekPlagiasi()` pada kelas `timTelaah` dan Method `cosine()` pada kelas `similarity`

| Input Pertama | Method dari kelas | Ouput Pertama / Input Kedua | Method dari kelas | Expected Result | Result | status |
|---|-------------------|-------------------------------|-------------------------|--|---|--------|
| Input abstrak pada <i>form</i> cek plagiasi | Tim telaah | Hasil pra proses teks abstrak | <code>similarity</code> | Sistem menampilkan persentase kemiripan antara abstrak yang dimasukkan oleh pengguna dengan data abstrak yang berada pada basis data sistem. | Sistem berhasil menampilkan persentase kemiripan antara abstrak yang dimasukkan oleh pengguna dengan data abstrak yang berada pada basis data sistem. | Valid |

Dari hasil pengujian pada Tabel 6.5 Hasil Pengujian Integrasi Method `cekPlagiasi()` pada kelas `timTelaah` dan Method `cosine()` pada kelas `similarity` dapat tervalidasi. Untuk tampilan interface dari hasil uji dapat dilihat pada Gambar 6.1.



Gambar 6.1 Tampilan hasil Pengujian integrasi method `cekPlagiasi()` dengan method `cosine()`.

2. Pengujian Integrasi Method `rekomendasiJuri()` pada kelas `admin` dan Method `countVi(calonJuri)` pada kelas `weightedProduct`

Method `rekomendasiJuri()` pada kelas `admin` merupakan method yang digunakan untuk mengolah data pelatihan-pelatihan karyawan yang telah mengikuti pelatihan juri pemula. Data tersebut nantinya akan diolah pada method `countVi(calonJuri)` yang terdapat pada kelas `weightedProduct` untuk mendapatkan rekomendasi juri tetap.

Tabel 6.6 Interaksi method `rekomendasiJuri()` dengan method `countVi()`

| No. | Method dari kelas <code>timtelaah</code> | Method dari kelas <code>similarity</code> | Goal |
|-----|--|---|--------------------------------------|
| 1. | <code>rekomendasiJuri()</code> | <code>countVi(calonJuri)</code> | Menghasilkan rekomendasi juri tetap. |

Prosedur uji yang dilakukan yaitu:

1. Menyiapkan kelas `timtelaah` untuk mengaktifkan method `rekomendasiJuri()`.
2. Menyiapkan input-input yang diperlukan method `rekomendasiJuri()`.
3. Mengeksekusi method `rekomendasiJuri()` dan method `countVi(calonJuri)`.

Hasil Pengujian Integrasi Method `rekomendasiJuri()` pada kelas admin dan Method `countVi(calonJuri)` pada kelas `weightedProduct` berdasarkan prosedur uji dapat dilihat pada Tabel 6.7.

Tabel 6.7 Hasil Pengujian Integrasi Method `rekomendasiJuri()` pada kelas admin dan Method `countVi()` pada kelas `weightedProduct`

| Input Pertama | Method dari kelas | Ouput Pertama / Input Kedua | Method dari kelas | Expepted Result | Result | status |
|-------------------------------|-------------------|---|-------------------|--|---|--------|
| Input data pelatihan karyawan | Admin | Hasil perhitungan jumlah pelatihan calon juri | Weighted Product | Sistem menampilkan rekomendasi juri tetap. | Sistem berhasil menampilkan rekomendasi juri tetap. | Valid |

Dari hasil pengujian pada Tabel 6.7 Hasil Pengujian Integrasi Method `rekomendasiJuri()` pada kelas admin dan Method `countVi(calonjuri)` pada kelas `weightedProduct` dapat tervalidasi. Untuk tampilan interface dari hasil uji dapat dilihat pada Gambar 6.2.



Gambar 6.2 Tampilan hasil Pengujian integrasi method `rekomendasiJuri()` dengan method `countVi(calonJuri)`.

6.3 Pengujian Validasi

Pengujian validasi dilakukan setelah tahap implementasi selesai. Pengujian validasi merupakan pengujian *blackbox* yang digunakan untuk mengetahui apakah sistem yang dibangun sudah memenuhi semua kebutuhan yang dibutuhkan serta untuk mengetahui apakah semua fungsi dari sistem yang dibangun bekerja sesuai dengan yang diharapkan. Dengan pengujian *blackbox* memungkinkan untuk

melakukan pengujian untuk menghasilkan sebuah kondisi yang diharapkan dengan suatu masukan untuk semua fungsi sistem.

6.3.1 Pengujian Validasi Login

- a. Kasus uji berhasil melakukan login.

Tabel 6.8 Kasus uji berhasil melakukan login

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji berhasil melakukan login. |
| Prosedur | <ol style="list-style-type: none"> 1. Mengisi kolom NPK. 2. Mengisi kolom password. 3. Menekan tombol login. |
| Hasil yang diharapkan | Menampilkan halaman utama user. |
| Hasil | Menampilkan halaman utama user. |
| Status | Valid. |

- b. Kasus uji gagal melakukan login dengan tidak mengisi kolom password.

Tabel 6.9 kasus uji gagal melakukan login dengan tidak mengisi kolom password

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji gagal melakukan login. |
| Prosedur | <ol style="list-style-type: none"> 1. Mengisi kolom NPK. 2. Tidak mengisi kolom password. 3. Menekan tombol login. |
| Hasil yang diharapkan | Menampilkan pesan “Harap isi bidang” pada kolom password. |
| Hasil | Menampilkan pesan “Harap isi bidang” pada kolom password. |
| Status | Valid. |

- c. Kasus uji gagal melakukan login dengan tidak mengisi kolom NPK.

Tabel 6.10 Kasus uji gagal melakukan login dengan tidak mengisi kolom NPK

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji gagal melakukan login. |
| Prosedur | <ol style="list-style-type: none"> 1. tidak mengisi kolom NPK. 2. Mengisi kolom password. 3. Menekan tombol login. |
| Hasil yang diharapkan | Menampilkan pesan “Harap isi bidang” pada kolom NPK. |

| | |
|---------------|--|
| Hasil | Menampilkan pesan “Harap isi bidang” pada kolom NPK. |
| Status | Valid. |

- d. Kasus uji gagal melakukan login dengan masukan yang tidak sesuai pada kolom NPK.

Tabel 6.11 Kasus uji gagal melakukan login dengan masukan yang tidak sesuai pada kolom NPK

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji gagal melakukan login admin. |
| Prosedur | <ol style="list-style-type: none"> 1. Mengisi kolom NPK dengan nilai yang tidak sesuai. 2. Mengisi kolom password. 3. Menekan tombol login. |
| Hasil yang diharapkan | Menampilkan pesan “Password/username yang anda masukkan salah”. |
| Hasil | Menampilkan pesan “Password/username yang anda masukkan salah”. |
| Status | Valid. |

- e. Kasus uji gagal melakukan login dengan masukan yang tidak sesuai pada kolom password.

Tabel 6.12 Kasus uji gagal melakukan login dengan masukan yang tidak sesuai pada kolom password

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji gagal melakukan login. |
| Prosedur | <ol style="list-style-type: none"> 1. Mengisi kolom NPK. 2. Mengisi kolom password dengan nilai yang tidak sesuai. 3. Menekan tombol login. |
| Hasil yang diharapkan | Menampilkan pesan “Password/username yang anda masukkan salah”. |
| Hasil | Menampilkan pesan “Password/username yang anda masukkan salah”. |
| Status | Valid. |

6.3.2 Pengujian Validasi Menilai makalah

a. Kasus uji berhasil melakukan penilaian.

Tabel 6.13 Kasus uji berhasil melakukan penilaian

| Nama Kasus Uji | Kasus uji berhasil melakukan penilaian. |
|------------------------------|--|
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai juri. 2. Memilih menu makalah. 3. Memilih tombol nilai pada kolom nilai sesuai dengan makalah yang ingin dinilai. 4. Memberi nilai pada setiap kategori penilaian yang ada dengan cara memilih nominal angka yang berada pada <i>form</i> penilaian. 5. Memilih tombol nilai. |
| Hasil yang diharapkan | Menampilkan halaman data makalah. |
| Hasil | Menampilkan halaman data makalah. |
| Status | Valid. |

b. Kasus uji gagal melakukan penilaian.

Tabel 6.14 Kasus uji gagal melakukan penilaian

| Nama Kasus Uji | Kasus uji gagal melakukan penilaian. |
|------------------------------|--|
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai juri. 2. Memilih menu makalah. 3. Memilih tombol nilai pada kolom nilai sesuai dengan makalah yang ingin dinilai. 4. Tidak memberi nilai tidak pada salah satu kategori penilaian yang ada pada <i>form</i> penilaian. 5. Memilih tombol nilai. |
| Hasil yang diharapkan | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Hasil | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Status | Valid. |

6.3.3 Pengujian Validasi Melihat data makalah *stream*

- a. Kasus uji berhasil melihat data makalah *stream*

Tabel 6.15 Kasus uji berhasil melihat data makalah *stream*

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji berhasil melihat data makalah <i>stream</i> . |
| Prosedur | 1. Login sebagai juri. 2. Memilih menu makalah. |
| Hasil yang diharapkan | Menampilkan data makalah sesuai <i>stream</i> . |
| Hasil | Menampilkan data makalah sesuai <i>stream</i> . |
| Status | Valid. |

6.3.4 Pengujian Validasi melihat hasil penilaian juri

- a. Kasus uji berhasil melihat penilaian juri

Tabel 6.16 Kasus uji berhasil melihat penilaian juri

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji berhasil melihat penilaian juri. |
| Prosedur | 1. Login sebagai juri. 2. Memilih menu makalah. |
| Hasil yang diharapkan | Menampilkan data makalah sesuai <i>stream</i> dengan kolom nilai berisi nilai juri pada makalah yang sudah dinilai sebelumnya, dan menampilkan tombol nilai untuk makalah yang belum dinilai. |
| Hasil | Menampilkan data makalah sesuai <i>stream</i> dengan kolom nilai berisi nilai juri pada makalah yang sudah dinilai sebelumnya, dan menampilkan tombol nilai untuk makalah yang belum dinilai. |
| Status | Valid. |

6.3.5 Pengujian Validasi Melihat hasil akhir

- a. Kasus uji berhasil melihat nilai akhir

Tabel 6.17 Kasus uji berhasil melihat nilai akhir

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil melihat nilai akhir |
| Prosedur | 1. Login sebagai juri. 2. Memilih menu Hasil Akhir. |
| Hasil yang diharapkan | Menampilkan data makalah sesuai <i>stream</i> dengan kolom nilai berisi nilai akhir hasil akumulasi semua nilai yang diberikan juri. |

| | |
|---------------|---|
| Hasil | Menampilkan data makalah sesuai stream dengan kolom nilai berisi nilai akhir hasil akumulasi semua nilai yang diberikan juri. |
| Status | Valid. |

6.3.6 Pengujian Validasi Mengunduh makalah

- a. Kasus uji berhasil mengunduh makalah

Tabel 6.18 Kasus uji berhasil mengunduh makalah

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji berhasil mengunduh makalah |
| Prosedur | 1. Login sebagai juri/tim telaah/admin. 2. Memilih menu makalah. |
| Hasil yang diharapkan | Makalah berhasil diunduh. |
| Hasil | Makalah berhasil diunduh. |
| Status | Valid. |

6.3.7 Pengujian Validasi Cek plagiasi

- a. Kasus uji berhasil melakukan cek plagiasi

Tabel 6.19 Kasus uji berhasil melakukan cek plagiasi

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil melakukan cek plagiasi |
| Prosedur | 1. Login sebagai tim telaah. 2. Memilih menu Cek Plagiasi. 3. Mengisi kolom abstrak dengan abstrak yang ingin diuji. |
| Hasil yang diharapkan | Menampilkan persentasi hasil kemiripan abstrak yang diuji dengan data abstrak yang ada pada basis data. |
| Hasil | Menampilkan persentasi hasil kemiripan abstrak yang diuji dengan data abstrak yang ada pada basis data. |
| Status | Valid. |

- b. Kasus uji gagal melakukan cek plagiasi

Tabel 6.20 Kasus uji gagal melakukan cek plagiasi

| | |
|-----------------------|---|
| Nama Kasus Uji | Kasus uji gagal melakukan cek plagiasi |
| Prosedur | 1. Login sebagai tim telaah. 2. Memilih menu Cek Plagiasi. |

| | |
|------------------------------|---|
| | 3. Tidak mengisi kolom abstrak dengan abstrak yang ingin diuji. |
| Hasil yang diharapkan | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Hasil | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Status | Valid. |

6.3.8 Pengujian Validasi Melihat makalah

- a. Kasus uji berhasil melihat makalah

Tabel 6.21 Kasus uji berhasil makalah

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil melihat makalah |
| Prosedur | 1. Login sebagai tim telaah. 2. Memilih menu makalah. |
| Hasil yang diharapkan | Menampilkan data makalah yang akan dikaji. |
| Hasil | Menampilkan data makalah yang akan dikaji. |
| Status | Valid. |

6.3.9 Pengujian Validasi Menentukan kelolosan makalah

- a. Kasus uji berhasil menentukan kelolosan makalah

Tabel 6.22 Kasus uji berhasil menentukan kelolosan makalah

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil menentukan kelolosan makalah |
| Prosedur | 1. Login sebagai tim telaah. 2. Memilih menu makalah. 3. Memilih tombol edit pada kolom edit makalah. 4. Mengisi kolom status pada <i>form</i> edit dengan memilih keterangan lolos / tidak lolos. 5. Memilih tombol update. |
| Hasil yang diharapkan | Status makalah berganti menjadi keterangan lolos / tidak lolos. |
| Hasil | Status makalah berganti menjadi keterangan lolos / tidak lolos. |
| Status | Valid. |

6.3.10 Pengujian Validasi Rekomendasi Juri

- a. Kasus uji berhasil memberi rekomendasi juri

Tabel 6.23 Kasus uji berhasil memberi rekomendasi juri

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji berhasil memberi rekomendasi juri. |
| Prosedur | 1. Login sebagai admin. 2. Memilih menu rekomendasi juri. 3. Memilih tombol rekomendasi juri. |
| Hasil yang diharapkan | Menampilkan hasil rekomendasi juri. |
| Hasil | Menampilkan hasil rekomendasi juri. |
| Status | Valid. |

6.3.11 Pengujian Validasi Melihat data calon juri

- a. Kasus uji berhasil melihat data calon juri

Tabel 6.24 Kasus uji berhasil melihat data calon juri

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil melihat data calon juri. |
| Prosedur | 1. Login sebagai admin. 2. Memilih menu rekomendasi juri. |
| Hasil yang diharapkan | Menampilkan data calon juri. |
| Hasil | Menampilkan data calon juri. |
| Status | Valid. |

6.3.12 Pengujian Validasi Melihat juri tetap

- a. Kasus uji berhasil melihat juri tetap

Tabel 6.25 Kasus uji berhasil melihat juri tetap

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil melihat juri tetap. |
| Prosedur | 3. Login sebagai admin. 4. Memilih menu juri. |
| Hasil yang diharapkan | Menampilkan data juri tetap. |
| Hasil | Menampilkan data juri tetap. |
| Status | Valid. |

6.3.13 Pengujian Validasi Menentukan *stream* juri

- a. Kasus uji berhasil menentukan *stream* juri

Tabel 6.26 Kasus uji berhasil menentukan *stream* juri

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil menentukan <i>stream</i> juri. |
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai admin. 2. Memilih menu juri. 3. Memilih tombol edit pada kolom edit juri. 4. Mengisi kolom <i>stream</i> pada <i>form</i> edit dengan memilih <i>stream</i> untuk juri. 5. Memilih tombol update. |
| Hasil yang diharapkan | <i>Stream</i> juri berhasil diupdate. |
| Hasil | <i>Stream</i> juri berhasil diupdate. |
| Status | Valid. |

6.3.14 Pengujian Validasi Memilih juri tetap

- a. Kasus uji berhasil memilih juri tetap

Tabel 6.27 Kasus uji berhasil memilih juri tetap

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil memilih juri tetap |
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai admin. 2. Memilih menu rekomendasi juri 3. Memilih tombol rekomendasi juri. 4. Memilih tombol tambah pada juri yang ingin ditetapkan sebagai juri tetap. |
| Hasil yang diharapkan | Menambah data juri tetap. |
| Hasil | Menambah data juri tetap. |
| Status | Valid. |

6.3.15 Pengujian Validasi menghapus juri tetap

- a. Kasus uji berhasil menghapus juri

Tabel 6.28 Kasus uji berhasil menghapus juri

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji berhasil menghapus juri. |
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai admin. 2. Memilih menu juri. 3. Memilih tombol hapus pada kolom hapus. |
| Hasil yang diharapkan | Juri tetap terhapus. |

| | |
|---------------|----------------------|
| Hasil | Juri tetap terhapus. |
| Status | Valid. |

6.3.16 Pengujian Validasi menentukan *stream* makalah

- a. Kasus uji berhasil menentukan *stream* makalah

Tabel 6.29 Kasus uji berhasil menentukan *stream* makalah

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji berhasil menentukan <i>stream</i> juri. |
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai admin. 2. Memilih menu makalah. 3. Memilih tombol edit pada kolom edit makalah. 4. Mengisi kolom <i>stream</i> pada <i>form</i> edit dengan memilih <i>stream</i> untuk makalah. 5. Memilih tombol update. |
| Hasil yang diharapkan | <i>Stream</i> makalah berhasil diupdate. |
| Hasil | <i>Stream</i> makalah berhasil diupdate. |
| Status | Valid. |

6.3.17 Pengujian Validasi Melihat data makalah terdaftar

- a. Kasus uji berhasil melihat data makalah terdaftar

Tabel 6.30 Kasus uji berhasil melihat data makalah terdaftar

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji berhasil melihat data makalah terdaftar. |
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai admin. 2. Memilih menu makalah terdaftar. |
| Hasil yang diharapkan | Menampilkan data makalah terdaftar. |
| Hasil | Menampilkan data makalah terdaftar. |
| Status | Valid. |

6.3.18 Pengujian Validasi Menentukan tim telaah makalah

- a. Kasus uji berhasil menentukan tim telaah makalah

Tabel 6.31 Kasus uji berhasil menentukan tim telaah makalah

| | |
|-----------------------|--|
| Nama Kasus Uji | Kasus uji berhasil menentukan tim telaah makalah. |
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai admin. 2. Memilih menu makalah terdaftar. 3. Memilih tombol edit pada kolom edit makalah. |

| | |
|------------------------------|---|
| | 4. Mengisi kolom tim telaah pada <i>form</i> edit dengan memilih nama tim telaah untuk mengkaji makalah. 5. Memilih tombol update. |
| Hasil yang diharapkan | Tim telaah makalah berhasil ditentukan. |
| Hasil | Tim telaah makalah berhasil ditentukan. |
| Status | Valid. |

6.3.19 Pengujian Validasi Melihat data makalah lolos

- a. Kasus uji berhasil melihat data makalah lolos

Tabel 6.32 Kasus uji berhasil melihat data makalah lolos

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji berhasil melihat data makalah lolos. |
| Prosedur | 3. Login sebagai admin. 4. Memilih menu makalah. |
| Hasil yang diharapkan | Menampilkan data makalah lolos. |
| Hasil | Menampilkan data makalah lolos. |
| Status | Valid. |

6.3.20 Pengujian Validasi Melihat data tim telaah

- a. Kasus uji berhasil melihat data tim telaah

Tabel 6.33 Kasus uji berhasil melihat data tim telaah

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil melihat data tim telaah |
| Prosedur | 1. Login sebagai admin. 2. Memilih menu tim telaah. |
| Hasil yang diharapkan | Menampilkan data tim telaah. |
| Hasil | Menampilkan data tim telaah. |
| Status | Valid. |

6.3.21 Pengujian Validasi Menambah data tim telaah

- a. Kasus uji berhasil menambah data tim telaah

Tabel 6.34 Kasus uji berhasil menambah data tim telaah

| | |
|-----------------------|--|
| Nama Kasus Uji | Kasus uji berhasil menambah data tim telaah |
| Prosedur | 1. Login sebagai admin. 2. Memilih menu tim telaah. |

| | |
|------------------------------|--|
| | 3. Memilih tombol tambah tim telaah. 4. Mengisi kolom npk. 5. Mengisi kolom password. 6. Memilih tombol tambah. |
| Hasil yang diharapkan | Menambah data tim telaah. |
| Hasil | Menambah data tim telaah. |
| Status | Valid. |

- b. Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom npk

Tabel 6.35 Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom npk

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom npk |
| Prosedur | 1. Login sebagai admin. 2. Memilih menu tim telaah. 3. Memilih tombol tambah tim telaah. 4. Tidak mengisi kolom npk. 5. Mengisi kolom password. 6. Memilih tombol tambah. |
| Hasil yang diharapkan | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Hasil | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Status | Valid. |

- c. Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom password

Tabel 6.36 Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom password

| | |
|-----------------------|---|
| Nama Kasus Uji | Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom password |
| Prosedur | 1. Login sebagai admin. 2. Memilih menu tim telaah. 3. Memilih tombol tambah tim telaah. 4. Mengisi kolom npk. 5. Tidak mengisi kolom password. |

| | |
|------------------------------|--|
| | 6. Memilih tombol tambah. |
| Hasil yang diharapkan | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Hasil | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Status | Valid. |

- d. Kasus uji gagal menambahkan data tim telaah dengan mengisi kolom npk dengan data yang sudah terdapat pada data tim telaah

Tabel 6.37 Kasus uji gagal menambahkan data tim telaah dengan mengisi kolom npk dengan data yang sudah terdapat pada data tim telaah

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji gagal menambahkan data tim telaah dengan mengisi kolom npk dengan data yang sudah terdapat pada data tim telaah |
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai admin. 2. Memilih menu tim telaah. 3. Memilih tombol tambah tim telaah. 4. Mengisi kolom npk dengan npk yang sudah terdapat pada data tim telaah. 5. Tidak mengisi kolom password. 6. Memilih tombol tambah. |
| Hasil yang diharapkan | Menampilkan pesan “Tim telaah telah terpilih” pada kolom yang kosong. |
| Hasil | Menampilkan pesan “Tim telaah telah terpilih” pada kolom yang kosong. |
| Status | Valid. |

6.3.22 Pengujian Validasi menghapus data tim telaah

- a. Kasus uji berhasil menghapus data tim telaah

Tabel 6.38 Kasus uji berhasil menghapus data tim telaah

| | |
|------------------------------|---|
| Nama Kasus Uji | Kasus uji berhasil menghapus data tim telaah. |
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai admin. 2. Memilih menu tim telaah. 3. Memilih tombol hapus pada kolom hapus. |
| Hasil yang diharapkan | Tim telaah tetap terhapus. |
| Hasil | Tim telaah terhapus. |

| | |
|---------------|--------|
| Status | Valid. |
|---------------|--------|

6.3.23 Pengujian Validasi Melihat data pelatihan calon juri

- a. Kasus uji berhasil melihat data pelatihan calon juri

Tabel 6.39 Kasus uji berhasil melihat data pelatihan calon juri

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil melihat data pelatihan calon juri |
| Prosedur | 1. Login sebagai admin. 2. Memilih menu rekomendasi juri. 3. Memilih tombol lihat pelatihan. |
| Hasil yang diharapkan | Menampilkan data pelatihan calon juri. |
| Hasil | Menampilkan data pelatihan calon juri. |
| Status | Valid. |

6.3.24 Pengujian Validasi Menambah pelatihan calon juri

- a. Kasus uji berhasil menambah pelatihan calon juri

Tabel 6.40 Kasus uji berhasil menambah pelatihan calon juri

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil menambah pelatihan calon juri |
| Prosedur | 1. Login sebagai admin. 2. Memilih menu rekomendasi juri. 3. Memilih tombol lihat pelatihan. 4. Memilih tombol tambah pelatihan. 5. Mengisi kolom peatihan. 6. Mengisi kolom tahun pelatihan. 7. Mengisi kolom tingkat pelatihan. 8. Memilih tombol tambah. |
| Hasil yang diharapkan | Menambah data pelatihan calon juri. |
| Hasil | Menambah data pelatihan calon juri. |
| Status | Valid. |

- b. Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom pelatihan

Tabel 6.41 Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom pelatihan

| | |
|-----------------------|---|
| Nama Kasus Uji | Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom pelatihan. |
| Prosedur | 1. Login sebagai admin. |

| | |
|------------------------------|--|
| | <ol style="list-style-type: none"> Memilih menu rekomendasi juri. Memilih tombol lihat pelatihan. Memilih tombol tambah pelatihan. Tidak mengisi kolom peatihan. Mengisi kolom tahun pelatihan. Mengisi kolom tingkat pelatihan. Memilih tombol tambah. |
| Hasil yang diharapkan | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Hasil | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Status | Valid. |

- c. Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom tahun pelatihan

Tabel 6.42 Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom tahun pelatihan

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom tahun pelatihan. |
| Prosedur | <ol style="list-style-type: none"> Login sebagai admin. Memilih menu rekomendasi juri. Memilih tombol lihat pelatihan. Memilih tombol tambah pelatihan. Mengisi kolom peatihan. Tidak mengisi kolom tahun pelatihan. Mengisi kolom tingkat pelatihan. Memilih tombol tambah. |
| Hasil yang diharapkan | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Hasil | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Status | Valid. |

- d. Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom tahun pelatihan

Tabel 6.43 Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom tahun pelatihan

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji gagal menambahkan data tim telaah dengan tidak mengisi kolom tahun pelatihan. |
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai admin. 2. Memilih menu rekomendasi juri. 3. Memilih tombol lihat pelatihan. 4. Memilih tombol tambah pelatihan. 5. Mengisi kolom peatihan. 6. Mengisi kolom tahun pelatihan. 7. Tidak mengisi kolom tingkat pelatihan. 8. Memilih tombol tambah. |
| Hasil yang diharapkan | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Hasil | Menampilkan pesan “Harap isi bidang” pada kolom yang kosong. |
| Status | Valid. |

6.3.25 Pengujian Validasi Logout

- a. Kasus uji berhasil logout

Tabel 6.44 Kasus uji berhasil logout

| | |
|------------------------------|--|
| Nama Kasus Uji | Kasus uji berhasil logout. |
| Prosedur | <ol style="list-style-type: none"> 1. Login sebagai admin/juri/tim telaah. 2. Memilih tombol foto profil pada kanan atas halaman. 3. Memilih tombol logout. |
| Hasil yang diharapkan | Menampilkan halaman utama. |
| Hasil | Menampilkan halaman utama. |
| Status | Valid. |

6.4 Pengujian Usabilitas

Pengujian usabilitas dilakukan setelah tahap implementasi telah selesai. Pada tahap ini dilakukan pengujian untuk mengetahui apakah sistem yang telah dibangun dapat digunakan dengan baik dan memberi kepuasan dan manfaat kepada pengguna dengan metode *WebQual* 4.0. Pada penelitian ini digunakan kuisiomer untuk melakukan pengujian usabilitas. Kuisiomer yang digunakan adalah *USE questionnaire* yang mana kuisiomer digunakan dengan cara memberikan penilaian menggunakan *skala likert* untuk meyakinkan responden menjawab dalam berbagai tingkatan pada setiap pertanyaan yang diberikan (Rahadi, 2014). Pertanyaan-pertanyaan yang digunakan adalah pertanyaan yang mengacu pada penelitian tiga area (dimensi) kualitas yang terdapat pada *WebQual* 4.0. pada penelitian ini digunakan kuisiomer yang diambil dari penelitian sebelumnya yang dilakukan oleh Qurrata Aynayya, Mochamad Chandra Saputra, dan Djoko Pramono dengan berjudul “Evaluasi *Usability* dan Rekomendasi Perbaikan Tampilan *Website* Seleksi Mahasiswa (SELMA) Universitas Brawijaya” (Aynayya, et al., 2018). Pada penelitian ini jumlah pertanyaan yang harus diisi oleh responden adalah 20 pertanyaan. Minimum responden dari kuisiomer pengujian usabilitas adalah 5 responden (Nielsen, 2012). Pada penelitian ini responden terdiri dari 5 karyawan pengelola PIA dan menggunakan 5 pilihan jawaban dengan setiap jawaban memiliki bobot masing-masing yang ditunjukkan pada Tabel 6.47.

Tabel 6.45 Bobot *skala Likert* untuk setiap pertanyaan

| No. | Keterangan | Bobot |
|-----|---------------------|-------|
| 1. | Sangat Tidak Setuju | 5 |
| 2. | Tidak Setuju | 4 |
| 3. | Netral | 3 |
| 4. | Setuju | 2 |
| 5. | Sangat Setuju | 1 |

6.4.1 Hasil Pengujian Usabilitas

Hasil Pengujian didapat setelah seluruh prosedur pengujian usabilitas dilakukan. Langkah selanjutnya adalah melakukan perhitungan dan analisis data hasil yang didapat dari pengisian kuisiomer yang dilakukan oleh 5 responden dengan 20 pertanyaan yang disajikan dalam bentuk *USE questionnaire*. Rekap hasil yang didapat dari responden ditunjukkan pada Tabel 6.49.

Tabel 6.46 Rekap Hasil Pengisian Kuisiomer

| Responden | Total Jawaban | | | | |
|-------------|---------------|----|---|---|----|
| | STS | TS | N | S | SS |
| Responden 1 | 0 | 0 | 0 | 0 | 20 |
| Responden 2 | 0 | 0 | 0 | 6 | 14 |

| | | | | | |
|-------------|---|---|---|----|----|
| Responden 3 | 0 | 0 | 1 | 2 | 17 |
| Responden 4 | 0 | 0 | 0 | 5 | 15 |
| Responden 5 | 0 | 0 | 1 | 1 | 18 |
| Total | 0 | 0 | 2 | 14 | 84 |

Berdasarkan rekap hasil pengisian kuisioner pada Tabel 6.49 maka dapat dihitung skor total dari penilaian hasil kuisioner yang dilakukan oleh 5 responden.

$$\text{Skor}_{\text{Total}} = (J_{\text{SS}} \times 5) + (J_{\text{S}} \times 4) + (J_{\text{N}} \times 3) + (J_{\text{TS}} \times 2) + (J_{\text{STS}} \times 1)$$

$$\text{Skor}_{\text{Total}} = (84 \times 5) + (14 \times 4) + (2 \times 3) + (0 \times 2) + (0 \times 2)$$

$$\text{Skor}_{\text{Total}} = 420 + 56 + 6 + 0 + 0$$

$$\text{Skor}_{\text{Total}} = 476$$

Berdasarkan skor total yang didapat, maka persentase skor untuk interpretasi hasil pengujian usabilitas dapat dihitung sebagai berikut:

$$\text{skor} = \frac{\text{Skor}_{\text{Total}}}{i \times r \times s} \times 100\%$$

$$\text{skor} = \frac{476}{20 \times 5 \times 5} \times 100\%$$

$$\text{skor} = 95,2\%$$

Berdasarkan hasil pengisian kuisioner yang ditunjukkan pada Tabel 6.49 terdapat 5 responden yang berpartisipasi dengan jumlah jawaban SS (Sangat Setuju) berjumlah 84, jawaban S (Setuju) berjumlah 14, jawaban N (Netral) berjumlah 2, jawaban TS (Tidak Setuju) berjumlah 0, serta jawaban STS (Sangat Tidak Setuju) berjumlah 0. Dari rekap hasil pengisian kuisioner tersebut maka dapat dihitung persentase skor yang pada perhitungan tersebut menghasilkan skor sebesar 95,2%.

6.5 Analisis Hasil Pengujian

Pada tahap ini dilakukan analisis dari hasil setiap pengujian yang telah dilakukan sebelumnya. Analisis hasil pengujian ini dilakukan untuk menarik kesimpulan bahwa hasil pengembangan sistem yang telah dilakukan telah berhasil dan memenuhi semua kebutuhan yang telah didefinisikan pada analisis kebutuhan.

6.5.1 Pengujian Unit

Pada pengujian unit dilakukan pengujian unit pada method `cekplagiasi()`, `rekomendasiJuri()`, dan `nilai()`. Pada pengujian unit dilakukan *basis path testing* yang menghasilkan *independent path* yang merupakan setiap jalur yang dapat dilewati dan dilakukan pengujian pada setiap method tersebut. Dari

independent path tersebut ditentukan kasus uji yang akan digunakan untuk pengujian unit. Hasil yang didapat pada pengujian unit berdasarkan kasus uji yang sudah ditentukan adalah valid pada setiap kasus uji nya yang berarti unit-unit sistem telah bekerja dengan baik.

6.5.2 Pengujian Integrasi

Pada pengujian Integrasi dilakukan pengujian pada method `cekplagiasi()` pada kelas `timTelaah` yang terintegrasi dengan method `cosine(a, b)` pada kelas `similarity` dan method `rekomendasiJuri()` pada kelas `admin` yang terintegrasi dengan method `countVi(calonJuri)` pada kelas `weightedProduct` yang mana pengujian dilakukan dengan menjalankan method-method tersebut kemudian membandingkan hasil yang didapat dengan hasil yang diinginkan dan sesuai dengan kebutuhan. Hasil dari pengujian integrasi ini menunjukkan keluaran yang sesuai dengan kebutuhan yang berarti integrasi antar method pada sistem telah bekerja dengan baik.

6.5.3 Pengujian Validasi

Pada pengujian validasi dilakukan pengujian pada semua fungsional yang terdapat pada sistem dengan cara menentukan setiap kemungkinan masukan yang dapat diberikan kepada sistem, kemudian menjalankan setiap kemungkinan masukan tersebut pada sistem yang telah dibangun. Hasil dari pengujian validasi ini adalah setiap kemungkinan masukan yang telah ditentukan menunjukan keluaran yang diinginkan sesuai dengan kebutuhan, sehingga memiliki status valid pada setiap kasus uji nya yang berarti sistem telah memenuhi kebutuhan-kebutuhan yang telah didefinisikan pada analisis kebutuhan.

6.5.4 Pengujian Usabilitas

Pada pengujian usabilitas dilakukan pengujian dengan memilih reponden sebanyak 5 orang yang terdiri dari karyawan-karyawan pengelola PIA yang kemudian diberi kuisioner untuk menentukan seberapa tepatnya sistem yang telah dibangun dengan kebutuhan yang dibutuhkan pada kegiatan PIA, serta mengukur kemudahan dalam penggunaan sistem. Pada pengujian usabilitas ini pengukuran dilakukan dengan menggunakan metode *webquel 4.0* serta menggunakan *USE questionnaire*. Hasil dari pengukuran tersebut didapat persentase skor sebesar 95,2% yang berarti masuk kedalam kategori baik, sehingga sistem sudah memenuhi kebutuhan.

BAB 7 PENUTUP

7.1 Kesimpulan

Berdasarkan proses-proses yang dilakukan dalam penelitian ini meliputi proses analisis kebutuhan, perancangan, implementasi, serta pengujian maka didapatkan hasil yang disimpulkan sebagai berikut:

1. Berdasarkan proses analisis kebutuhan yang telah dilakukan maka mendapatkan hasil berupa 25 kebutuhan fungsional serta 1 kebutuhan non fungsional yang dibutuhkan untuk mengelola kegiatan penjurian PIA secara online. Selain itu didapat pula hasil berupa pemodelan kebutuhan yang dapat digunakan sebagai pedoman dalam proses perancangan yaitu berupa *use case diagram* dan *use case scenario*. Pada proses perancangan yang dilakukan berdasarkan analisis kebutuhan didapatkan hasil berupa perancangan arsitektur yang berupa rancangan arsitektur MVC, perancangan *class diagram*, serta perancangan *sequence diagram* yang menjelaskan urutan proses dari setiap fungsionalitas sistem, perancangan data yang menghasilkan perancangan basis data berupa ERD (*Entity relationship diagram*), serta perancangan algoritma berupa rancangan aloritme-algoritme yang akan digunakan dalam proses implementasi. Pada proses perancangan antarmuka dihasilkan gambaran antarmuka dari sistem yang akan dibangun. Serta pada proses implementasi menghasilkan implementasi basis data berupa PDM (*Physical Data Model*), implelementasi kode program, serta implementasi antarmuka sistem yang sesuai dengan perancangan.
2. Berdasarkan hasil pengujian yang telah dilakukan sistem dapat membantu tim telaah dalam menentukan persentase plagiasi suatu abstrak makalah dengan menggunakan algoritme *cosine similarity*.
3. Berdasarkan hasil pengujian yang telah dilakukan sistem dapat memberikan rekomendasi juri tetap dengan menggunakan algoritme *weighted product*.
4. Berdasarkan hasil pengujian usabilitas mengenai peran sistem dalam mempercepat dan mempermudah proses penjurian online didapat skor sebesar 95,2%, sehingga dapat disimpulkan bahwa peran sistem dalam mempercepat dan memudahkan kegiatan penjurian ini sudah terpenuhi.

7.2 Saran

Saran yang diberikan berdasarkan hasil penelitian untuk pengembangan sistem penjurian online pengelola kegiatan PIA ini adalah sebagai berikut:

1. Dapat menyimpan riwayat penilaian pada form penilaian suatu makalah.
2. Terdapat fungsi submit nilai makalah dari juri ke admin.
3. Dapat menampilkan lebih dari 1 *role* user dalam 1 kali login.

DAFTAR PUSTAKA

Alamanda, R., Suhery, C. & Brianorman, Y., 2016. Aplikasi Pendeteksi Plagiat Terhadap Karya Tulis Berbasis Web Menggunakan Natural Language Processing dan Algoritma Knut-Morris-Pratt. *Jurnal Coding, Sistem Komputer Untan*, Volume 04, pp. 33-44.

Arshad, 2011. *Perancangan Sistem Informasi Pariwisata Berbasis Web di Dinas Kebudayaan dan Pariwisata Kabupaten Muna*, s.l.: s.n.

Asian, J., 2007. *RMIT University*. [Online] Available at: researchbank.rmit.edu.au [Diakses 15 November 11 2017].

Astuti, D., Pinandito, A. & Dewi, R. K., 2017. Sistem Rekomendasi Lowongan Pekerjaan Untuk Fresh raduate Menggunakan Metode Weighted Product Berbasis Android. *Jurnal Pengembangan Tknoogi Informasi dan Ilmu Komputer*, Volume 12, pp. 1518-1525.

Aynayya, Q., Saputra, M. C. & Pramono, D., 2018. Evaluasi Usability da Rekomendasi Perbaikan Tamplan Website Seleksi Mahasiswa (SELMA) Universitas Brawijaya. *Jurnal Pengembangan Teknologi dan Ilmu Komputer*, Volume 2, pp. 1446-1456.

Dewanto, I. J., 2004. System Development Life Cycle dengan Beberapa Pendekatan. *Jurnal FASILKOM*, Volume 2.

Hepyan, C. L., Gunadhi, E. & Kurniawati, R., 2012. Rekayasa Perangkat Lunak aplikasi Penjualan Obat pada Apotek Mulya Abadi. *ISSN*, Volume 9, pp. 2302-7339.

Imbar, R. V., A., Ayub, M. & Rehatta, A., 2014. Implementasi Cosine Similarity dan Algoritma Smith-Waterman untuk Mendeteksi Kemiripan Teks. *Jurnal Informatika*, Volume 10, pp. 31-42.

Konchady, M., 2006. *Text Mining Application Programming..* Boston: Charles River Media.

Lukmanul, H., 2010. *Bikin website Super Keren dengan PHP dan jQuery*. Yogyakarta: Lokomedia.

Mustaqbal, M. S., Firdaus, F. R. & Rahmadi, H., 2015. Penujian Aplikasi Menggunakan Black Box Testing Boundary Value Analysis (Studi kasus: Aplikasi Prediksi Kelulusan SNMPTN). *ISSN*, pp. 2407-3911.

Nielsen, J., 2012. *Nielsen Norman Group*. [Online] Available at: <https://www.nngroup.com/articles/how-many-test-users/> [Diakses februari 2018].

Nofriansyah, D., 2015. *Konsep Data Mining Vs Sistem Pendukung Keputusan*. Sleman: Deepublish.

Noviyanto, A., 2010. *Burn Your PHP Code Using Codeigniter*. Yogyakarta: Skripta Media Crative.

Prabowo, D., 2015. website E-Commerce Menggunakan model View Controller (MVC) dengan Framework CodeIgniter. *Jurnal Ilmiah DASI*, Volume 16, pp. 23-29.

Pratama, D. I., 2017. *Menenal Pupuk Kaltim Inovation Award* [Wawancara] (21 8 2017).

Pressman, R. S., 2009. *Software Enginnering: A Practitioner's Approach*. 7 penyunt. New York: Mc Graw-Hill.

Prisdiana, O., Aknuranda, I. & Pradana, F., 2018. Pengembangan Sistem Reservasi Penginapan untuk Youth Hostel Griya Brawijaya. *Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer*, Volume 2, pp. 1611 - 1619.

Rahadi, D. R., 2014. Pengukuran Usability Sistem Menggunakan Use Questionnaire Pada Aplikasi. *Jurnal Sistem Informasi (JSI)*, Volume 6, pp. 661-671.

Sanjaya, I., 2012. Pengukuran Kualitas Layanan Website Kementrian KOMINFO dengan Menggunakan Metode WEBUAL 4.0. *Jurnal Penelitian IPTEK-KOM*, Volume 14.

Sidik, B., 2006. *Pemrograman Web dengan PHP*. Bandung: Informatika.

Weiss, S., Indurkha, N., zhang, T. & Damerau, F., 2005. *Text Mining: Predictive Method for Analysing Unstructured information*. New York: Springer.

